

A  
Mini Project  
On  
**APPLICATION AND EVALUATION OF A K-MEDOIDS BASED SHAPE  
CLUSTERING METHOD FOR AN ARTICULATED DESIGN SPACE**

(Submitted in partial fulfilment of the requirements for the award of Degree)

**BACHELOR OF TECHNOLOGY**  
In  
**COMPUTER SCIENCE AND ENGINEERING**  
By

**SANIYA (217R1A0551)**  
**T.RAGHUVARAN (217R1A0557)**  
**R.HARSHAVARDHAN (217R1A0548)**

Under the Guidance of  
**Dr. V. NARESH KUMAR**  
(Associate Professor, HOD CSE-II)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**CMR TECHNICAL CAMPUS**  
**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct. 1956,

Kandlakoya(V), Medchal Road, Hyderabad-501401.

**2021-2025**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



### CERTIFICATE

This is to certify that the project entitled “APPLICATION AND EVALUATION OF A K-MEDOIDS BASED SHAPE CLUSTERING METHOD FOR AN ARTICULATED DESIGN SPACE” being submitted by SANIYA (217R1A0551), T.RAGHUVARAN (217R1A0557), R.HARSHAVARDHAN (217R1A0548) in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2024-25

The results embodied in this thesis have not been submitted to any other University Institute for the award of any degree or diploma.

**Dr. V. NARESH KUMAR**  
Associate Professor, HOD CSE-II  
INTERNAL GUIDE

**Dr. A. RAJI REDDY**  
DIRECTOR

**Dr. NUTHANAKANTI BHASKAR**  
HOD

**EXTERNAL EXAMINER**

Submitted for viva voice Examination held on \_\_\_\_\_

## ACKNOWLEDGEMENT

A part from the efforts of us, the success of any project depends largely on the Encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

**Dr. V. Naresh kumar**, Associate Professor, HOD CSE-II for his exemplary guidance, monitoring And constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Dr. K. Maheswari, Dr. J. Narasimha Rao, Mrs. K. Shilpa, Mr. K. Ranjith Reddy.** for their cordial support, valuable information and guidance, which helped us in Completing this task through various stages.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the Course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course Of this project.

We are also thankful to **Dr. Nuthanakanti Bhaskar** , Head of Department of Computer Science And Engineering for providing encouragement and support for completing this project successfully.

The guidance and support received from all the members of CMR Technical Campus who contributed to the completion of the project we are their constant support and help.

Finally, we would like to this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project

**SANIYA(217R1A0551)**

**T.RAGHUVARAN(217R1A0557)**

**R.HARSHAVARDHAN(217R1A0548)**

## **ABSTRACT**

Research on articulating the design space in computational generative systems is ongoing, to overcome the issue of possible overwhelming multiplicity and redundancy of emerging design options. The project contributes to this line of research of design space articulation, in order to facilitate designers' successful exploration in computational design. We have recently developed a method for shape clustering using K-Medoids, a machine learning-based strategy. The method performs clustering of similar design shapes and retrieves a representative shape for each cluster in 2D grid-based representation. In this project, we present a progress in our project where the method has been applied to a new test case, and empirically verified using clustering evaluation methods. Our clustering evaluation results show comparable accuracy when assessed against an external study and provide insight into the evaluation criteria for machine learning methods, as presented in the project.

## LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
Figure 3.1	Project Architecture	7
Figure 3.2	The flowchart of the SC-KM Method	9
Figure 3.3	Applying the SC-KM method to a dataset of 100 Shapes	9
Figure 3.4	Use Case Diagram	10
Figure 3.5	Class Diagram	11
Figure 3.6	Sequence Diagram	12
Figure 3.7	Activity Diagram	13

## **LIST OF RESULTS**

<b>RESULTS NO</b>	<b>RESULTS NAME</b>	<b>PAGE NO</b>
RESULT 5.1	Upload Dataset	23
RESULT 5.2	Upload Shapes folder	23
RESULT 5.3	Preprocess & Hamming Distance Calculation	24
RESULT 5.4	Shapes are processed and saved in application memory	24
RESULT 5.5	clustering process	25
RESULT 5.6	Triangle and Star shapes in Cluster 1 Circle shapes in Cluster 2	25
RESULT 5.7	Rectangle shapes in Cluster 3 & 4	26
RESULT 5.8	F contains different shapes	26

# TABLE OF CONTENT

<b>ABSTRACT</b>	i
<b>LIST OF FIGURES</b>	ii
<b>LIST OF RESULTSS</b>	iii
<b>1. INTRODUCTION</b>	1
1.1 PROJECT SCOPE	1
1.2 PROJECT PURPOSE	1
1.3 PROJECT FEATURES	1
<b>2. SYSTEM ANALYSIS</b>	2
2.1 PROBLEM DEFINITION	2
2.2 EXISTING SYSTEM	3
2.2.1 LIMITATIONS OF THE EXISTING SYSTEM	3
2.3 PROPOSED SYSTEM	4
2.3.1 ADVANTAGES OF PROPOSED SYSTEM	4
2.4 FEASIBILITY STUDY	5
2.4.1 ECONOMIC FESIBILITY	5
2.4.2 TECHNICAL FEASIBILITY	5
2.4.3 SOCIAL FEASIBILITY	5
2.5 HARDWARE & SOFTWARE REQUIREMENTS	6
2.5.1 HARDWARE REQUIREMENTS	6
2.5.2 SOFTWARE REQUIREMENTS	6
<b>3. ARCHITECTURE</b>	7
3.1 PROJECT ARCHITECTURE	7
3.2 DESCRIPTION	7
3.3 USE CASE DIAGRAM	10
3.4 CLASS DIAGRAM	11
3.5 SEQUENCE DIAGRAM	12
3.6 ACTIVITY DIAGRAM	13
<b>4. IMPLEMENTATION</b>	14
4.1 SOURCE CODE	14
<b>5. RESULTS</b>	23
<b>6. TESTING</b>	27
6.1 INTRODUCTION TO TESTING	27
6.2 TYPES OF TESTING	27
6.2.1 UNIT TESTING	27
6.2.2 INTEGRATION TESTING	27
6.2.3 FUNCTIONAL TESTING	28
6.3 TEST CASES	28

6.3.1	UPLOADING DATASET	33
6.3.2	PROCESSING DATASET	29
<b>7.</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>30</b>
7.1	PROJECT CONCLUSION	30
7.2	FUTURE SCOPE	31
<b>8.</b>	<b>BIBLIOGRAPHY</b>	<b>32</b>
8.1	REFERENCES	32
8.2	WEBSITES	33



# **1. INTRODUCTION**

# 1. INTRODUCTION

## 1.1 PROJECT SCOPE

This project, titled “**Application and Evaluation of a K-Medoids Based Shape Clustering Method for an Articulated Design Space**” focuses on the development of a Generative Design System (GDS) that integrates an architectural shape clustering mechanism into the design search process. Generative Design Systems (GDSs) are the computational frameworks that involve parametric modelling and form finding, aimed at identifying the most successful or best candidates within a generated set of designs. However, existing GDSs often face challenges due to the production of an excessive number of designs, many of which may have redundant characteristics. This research addresses these challenges by incorporating a machine learning-based clustering mechanism, specifically using the K-Medoids algorithm, to organize the design space effectively.

## 1.2 PROJECT PURPOSE

The purpose of this project is to enhance the generative design process by enabling more effective navigation and interaction between designers and computational systems. Traditional GDSs lack established organizational mechanisms, which makes it difficult for designers to cope with the vast number of generated designs. By organizing similar designs into clusters and highlighting representative designs, this project aims to support successful human-computer interaction and facilitate the exploration of a diverse design space. The integration of shape comparison and similarity/difference finding methods into the GDS allows for more efficient and targeted design exploration.

## 1.3 PROJECT FEATURES

Key features of this project include the application and further development of the **Shape Clustering using K-Medoids (SC-KM)** method. The project involves applying the SC-KM method to a new dataset of architectural shapes, adding an algorithm to convert boundary-based building design shapes into grid-based shapes for comparison, and utilizing clustering evaluation metrics to assess the clustering results. The project also emphasizes the importance of evaluating machine learning strategies, particularly unsupervised methods like clustering, to ensure the effectiveness of the proposed system. This evaluation is crucial for advancing the integration of artificial intelligence methods into computational design processes.

## **2. SYSTEM ANALYSIS**

## **2.SYSTEM ANALYSIS**

### **SYSTEM ANALYSIS**

System Analysis is a vital phase in the development of the "Application and Evaluation of a K-Medoids Based Shape Clustering Method for an Articulated Design Space." In this stage, existing generative design systems (GDSs) are examined to understand their operational dynamics and challenges. The system analyst acts as an interrogator, exploring the intricacies of current systems, including design generation methods and evaluation processes. A central question guiding this analysis is, "What must be done to improve the articulation of the design space and enhance GDS usability?" This examination identifies critical inputs, such as architectural design data and clustering algorithms, while also analyzing the relationships among various components.

Additionally, the analysis focuses on the overwhelming volume of designs produced by existing systems, which can hinder effective exploration. By understanding the current limitations and the nature of the design data, the analyst can formulate strategies for implementing K-Medoids based shape clustering to facilitate better organization and navigation within the design space. Upon completion of this analysis, the analyst will gain a robust understanding of the adjustments needed to develop an effective generative design system, ultimately aiding users in efficiently identifying and selecting preferred design options.

### **2.1 PROBLEM DEFINITION**

A detailed study of the generative design process must be conducted using techniques such as shape clustering and clustering evaluation metrics. The data generated by these techniques must be analyzed to understand how current Generative Design Systems (GDSs) function, forming the basis of the existing system. This existing system undergoes a close examination to identify problem areas, such as the overwhelming number of similar designs and the lack of effective organizational mechanisms. The designer functions as a problem solver, proposing solutions that integrate a K-Medoids based shape clustering method to articulate the design space. These solutions are then evaluated against the existing system analytically, and the most effective proposal is selected. The proposal is presented to the user for endorsement and may be refined based on feedback. This iterative process

continues until the user is satisfied with the final solution, ensuring it effectively addresses the identified issues.

## **2.2 EXISTING SYSTEM**

In the existing system based on Articulating design solutions in computational generative schemes has gained attention recently (Rodrigues et al., 2017; Brown & Mueller, 2019). Effective exploration of design problems requires a clear representation of the geometries involved (Turrin et al., 2016). Achieving diversity in design alternatives is crucial to avoid repetition and enhance the generative process, ensuring the outcomes remain engaging for designers (Rodrigues et al., 2017).

Current methods for organizing big data primarily include classification and clustering. Clustering, an unsupervised learning technique, identifies structures within unlabelled data (Jain et al., 1999; Velmurugan & Santhanam, 2010) and is advantageous for uncovering hidden patterns (Han et al., 2011). However, its integration into Generative Design Systems (GDSs) is still limited. Notable studies, such as that of Rodrigues et al., emphasize the need for further exploration of clustering algorithms for architectural layouts. Additionally, tools like the Ivy tool for Grasshopper have attempted to incorporate K-Means clustering, primarily focusing on mesh segmentation. The research of Jayanti et al. (2009) represents early efforts in managing CAD repositories and encourages further investigation into suitable clustering algorithms for shape similarity assessments.

### **2.2.1 LIMITATIONS OF EXISTING SYSTEM**

- Ineffective Clustering Methods
- Lack of Organizational Frameworks
- Less accuracy
- Low Efficiency

To avoid all these limitations and make the working more accurately the system needs to be implemented efficiently.

## 2.3 PROPOSED SYSTEM

The methodology of this work involved an extensive literature study, experimenting and prototyping, and testing and validation. These methods have led to the development of a new shape clustering method, the (SC-KM). Developing the method, the protocol included employing a grid-based descriptor, formulating a shape difference finding method, and implementing the K-Medoids clustering algorithm. In our previous projects, the SC-KM method was fully described (Yousif & Yan, 2019a, b). Experimenting and prototyping were performed in the Rhino/Grasshopper R environment. For the grid-based shape generation and description, modelling was pursued, using visual programming, in addition to customized programs written in Grasshopper R Python, and C# languages. For the SC-KM, a package of algorithmic set was developed, primarily using the GH C Python tool (Abdel Rahman, 2017) that allows communicating with the Python environment, and incorporating the scientific libraries and modules. For explaining the test-case application and evaluation, the focus of this project, there is a need to concisely describe the SC-KM method and introduce its algorithms.

For shape description, a typical grid-based approach was employed to define the shape characteristics. The shape difference finding method we developed started with investigating the distance-based diversity measure (Toffolo & Benini, 2003). Since the dataset in our case is a set of shapes, we needed a method applied to multidimensional space such as the architectural design space. As such, we formulated two sets of algorithms: (1) pair-wise shape difference and the Hungarian algorithm, and (2) K-Medoids clustering.

### 2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- Enhanced Design Diversity.
- Efficient Shape Comparison.
- Improved Design Organization
- High accuracy
- High efficiency

## **2.4 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

### **2.4.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **2.4.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### **2.4.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the

users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## **2.5 HARDWARE & SOFTWARE REQUIREMENTS**

### **2.5.1 HARDWARE REQUIREMENTS:**

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- System : Pentium IV 2.4 GHz or Greater
- Hard Disk : 40 GB or more
- Ram : 512 MB or more

### **2.5.2 SOFTWARE REQUIREMENTS:**

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

- Operating system : Windows 7 Professional or Later
- Coding Language : python



# **3. ARCHITECTURE**

### 3.1 PROJECT ARCHITECTURE

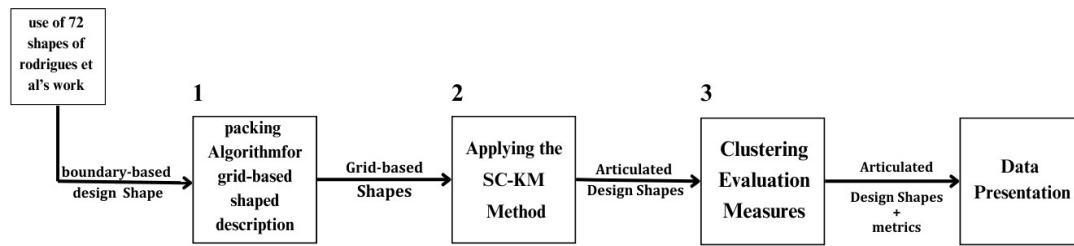


Figure 3.1 : Project Architecture of Application and evaluation of a K-Medoids based shape clustering method for an articulated design space

### 3.2 DESCRIPTION

**Boundary-based design shape :** Boundary-based shape representation is a method of representing the shape of an object by using only the outer boundary of the shape.

**Packing Algorithm for Grid-based Shaped description :** These packing algorithms can be used to optimize the layout of these shapes within a given space. This could involve:

- **Efficient Space Utilization:** Ensuring that the grid-based representations of different architectural components are packed tightly, minimizing wasted space.
- **Design Space Reduction:** By effectively packing shapes, you can reduce the overall design space that needs to be searched, which can improve computational efficiency.
- **Cluster Optimization:** If your design clusters are represented as grid-based shapes, packing algorithms can help in organizing these clusters within a defined area or volume

**Grid-Based shapes :** Grid shapes are parametric shapes that contain a grid of horizontal and vertical lines, and may also include an optional clipping frame. They can be used as building blocks for more complex shapes.

**SC-KM Method :** The Shape Clustering using K-Medoids (SC-KM) Method is an advanced technique that adapts the traditional K-Medoids clustering algorithm to focus on clustering shapes, specifically in the context of architectural design spaces. The method is particularly useful when dealing with complex shapes.

**Articulated Design Space :** An articulated design space is a structured and multi-dimensional environment where design variables, constraints, and configurations are systematically organized for exploration and optimization. It allows for detailed analysis and manipulation of design elements, facilitating informed and innovative decisions in complex design processes such as architecture and engineering.

**Clustering Evaluation Measures :** Clustering evaluation measures are metrics used to assess the quality of clustering results by determining how well the clusters reflect the underlying structure of the data. These measures can be internal, evaluating the compactness and separation of clusters, external, comparing the clustering results to known ground truth labels, or relative, identifying the optimal number of clusters. They are crucial for validating the effectiveness of clustering algorithms and ensuring meaningful groupings of data.

**Metrics :** Metrics refer to quantitative measures used to assess, compare, and evaluate the performance, quality, or characteristics of various processes, systems, or outcomes. In the context of clustering, metrics specifically help determine the effectiveness of clustering algorithms by measuring aspects such as cluster cohesion, separation, and accuracy.

**Data presentation :** It is crucial for enhancing understanding of clustering results and illustrating relationships among architectural shapes. It facilitates evaluation and validation of the clustering quality through visualizations and metrics, supports effective communication of findings to diverse audiences, and provides actionable insights for informed decision-making in design processes.

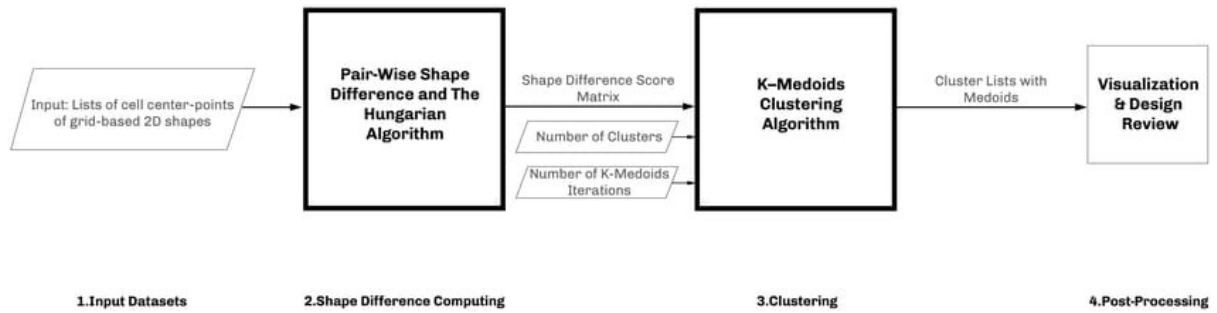


Figure 3.2 : The flowchart of the SC-KM Method, comprising of the pair-wise shape difference finding and the Hungarian algorithm, the k-Medoids clustering, and additional input dataset, and data processing nodes.

The dataset in our case is a set of shapes, we needed a method applied to multidimensional space such as the architectural design space. As such, we formulated two sets of algorithms: (1) pair-wise shape difference and the Hungarian algorithm, and (2) K-Medoids clustering, depicted as a flowchart in Fig 3.2.

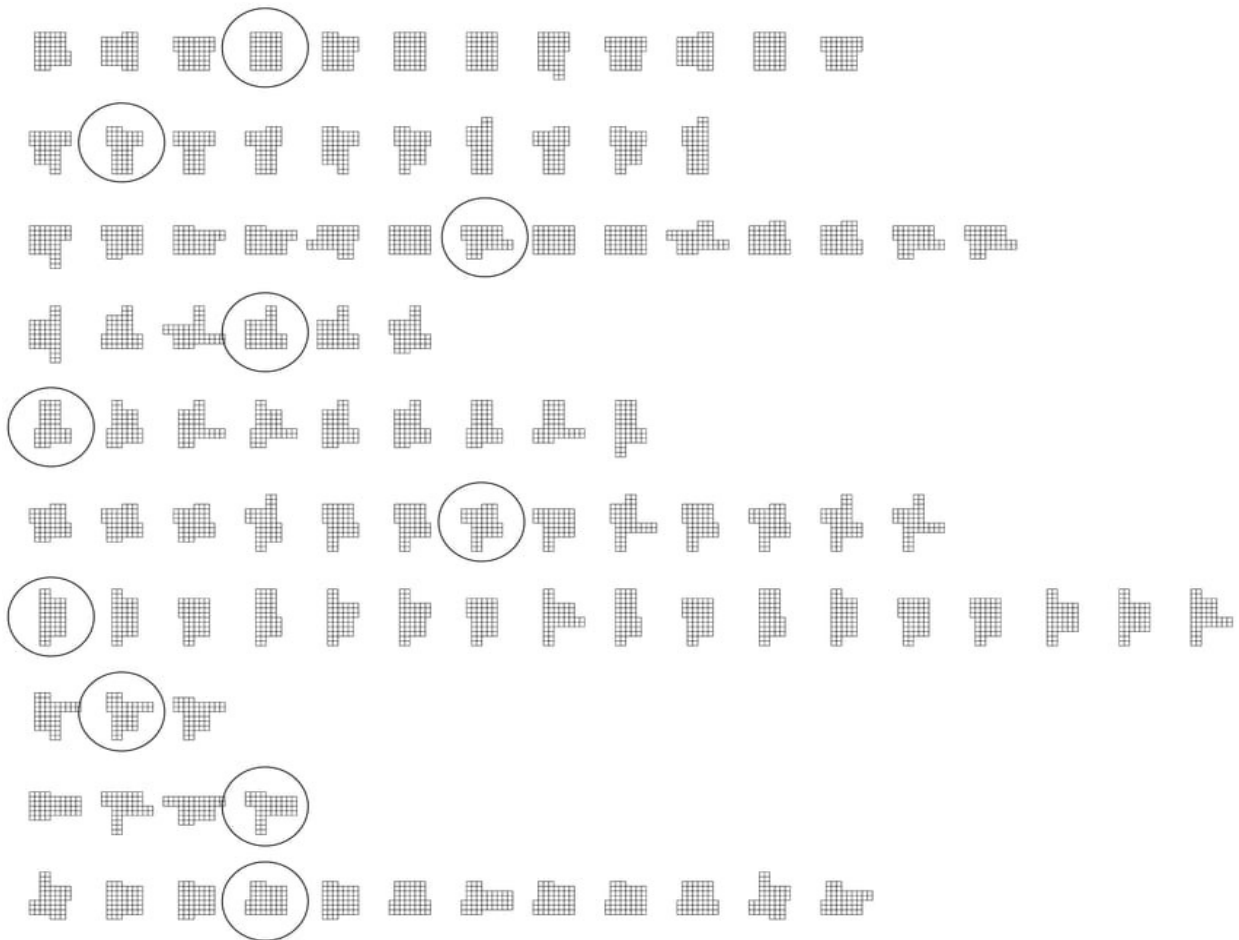
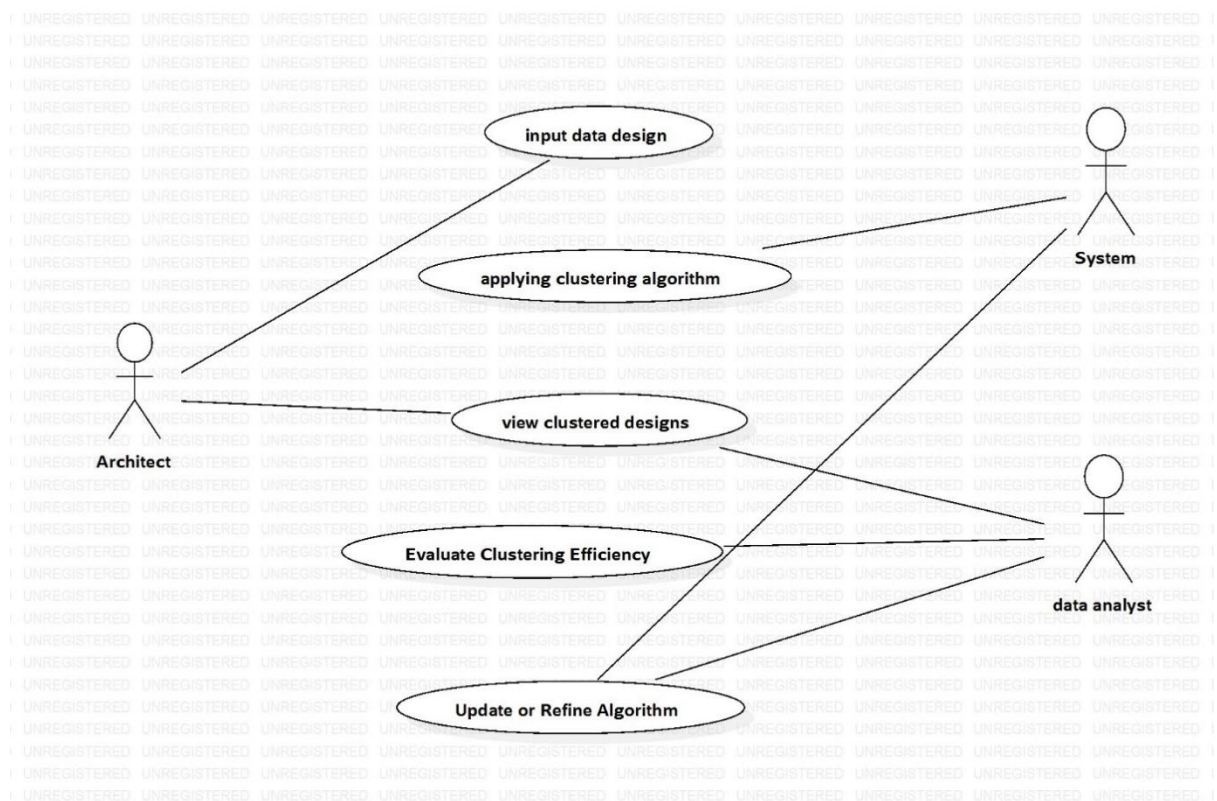


Figure 3.3 : Applying the SC-KM method to a dataset of 100 Shapes, Clustered into 10 rows of clusters with each cluster's representative circled

In multiple test cases, the SC-KM method showed successful clustering results, when evaluated according to perceptual coherence—visual examination of each cluster’s coherence. Figure 2 depicts one of those cases, in which 100 shapes of architectural typological designs of 48-cell grid-based representation were clustered into 10 clusters, and 10 representative shapes are highlighted (circled)

### 3.3 USE CASE DIAGRAM

In the use case diagram we have basically three actors: the Architect, the Data Analyst, and the System. The Architect has the rights to log in, input design data, and view clustered designs. The Data Analyst also has the ability to log in and access resources, allowing them to evaluate clustering efficiency, view the clustered designs, and update or refine the algorithm based on their analysis. The System acts as the central component, applying the clustering algorithm and facilitating the interactions between the Architect and Data Analyst. This structure defines the roles and responsibilities of each actor in the system, showcasing their interactions with the key use cases within the diagram.

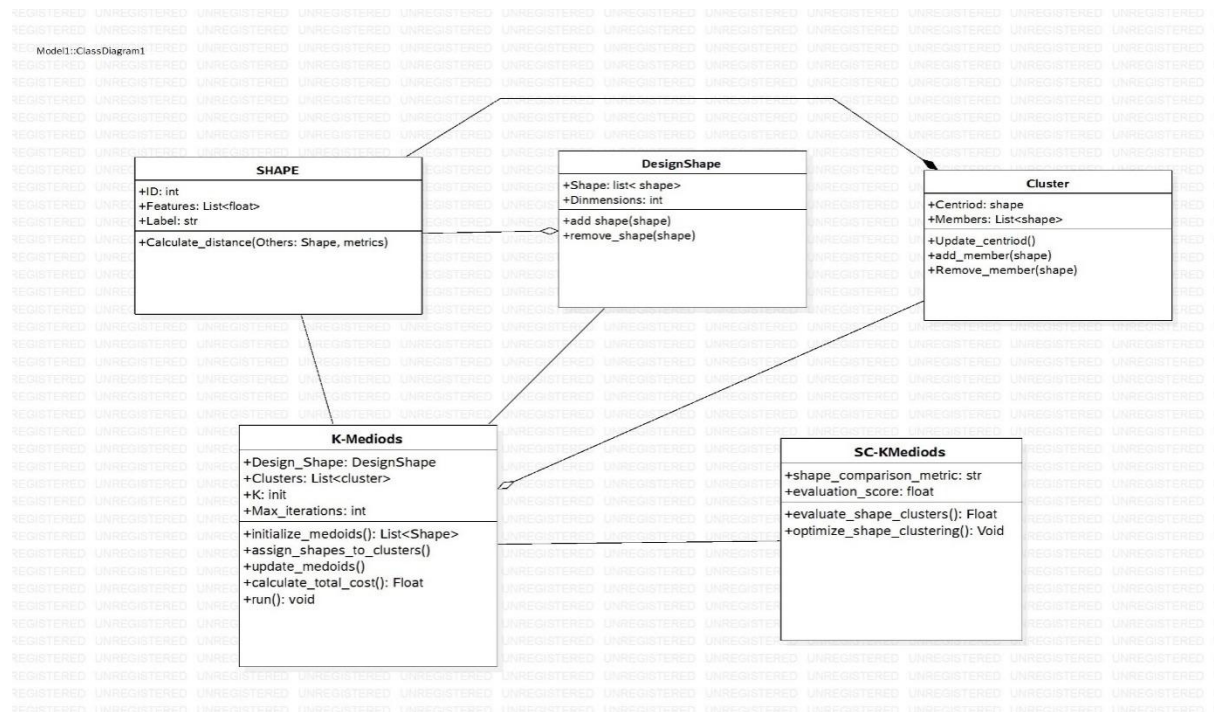


**Figure 3.4 :** Use Case Diagram for Application and evaluation of a K-Medoids based shape clustering method for an articulated design space

### 3.4 CLASS DIAGRAM

The **Shape** class represents individual shapes, featuring attributes such as a unique identifier (id), a list of numerical features, and an optional evaluation label. It includes a method for calculating distances to other shapes using metrics like Euclidean. The **Design Space** class holds a collection of shapes with attributes for the shape list and the space's dimension, providing methods to add or remove shapes while aggregating multiple Shape instances in a one-to-many relationship. The **Cluster** class defines clusters with a centroid (medoid) and a list of members, allowing for centroid updates and member management, also maintaining a one-to-many relationship with Shape.

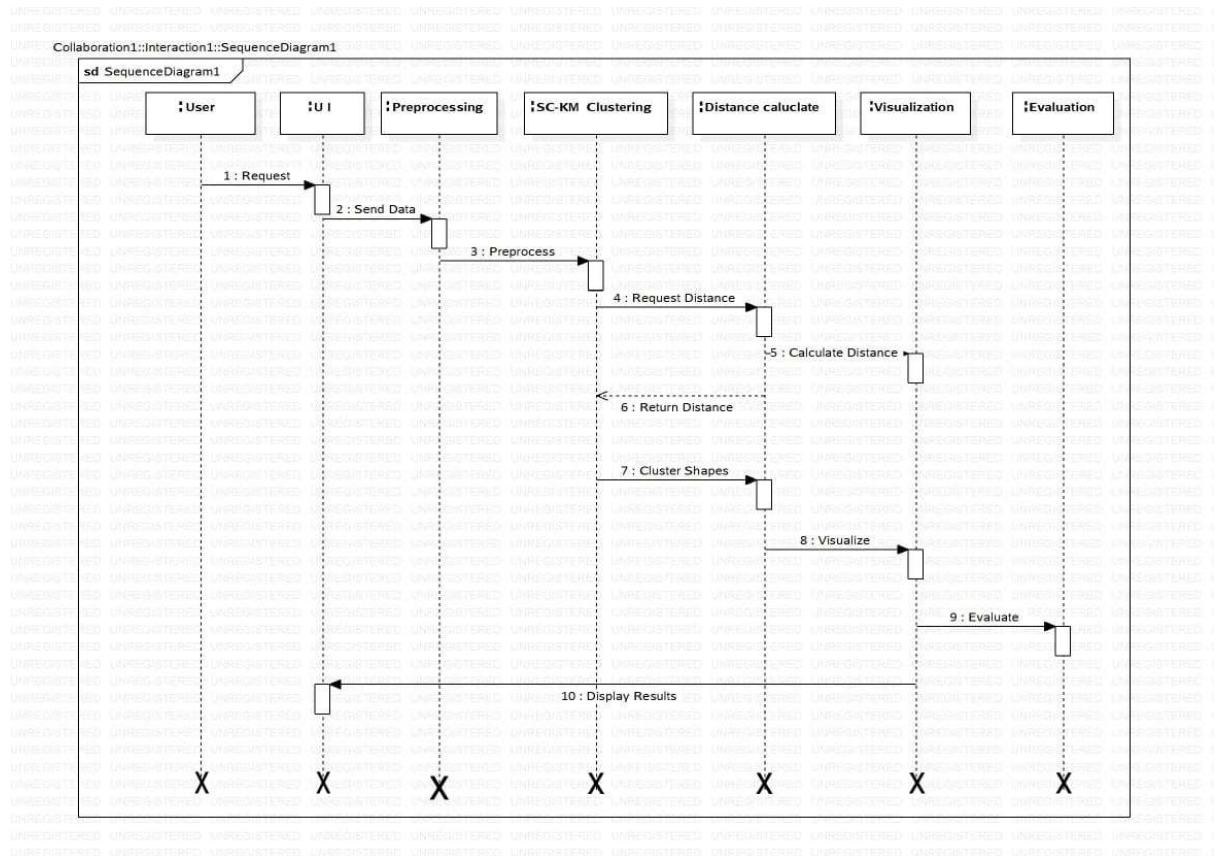
The **K-Medoids** class coordinates the clustering process, containing attributes like the design space, a list of clusters, the desired number of clusters (k), and maximum iterations. It includes methods for initializing medoids, assigning shapes, and calculating total cost, aggregating multiple Cluster instances. Lastly, the **SC-K-Medoids** class enhances K-Medoids by incorporating a shape comparison metric and an evaluation score, featuring methods for cluster evaluation and optimization while leveraging K-Medoids for shape-specific clustering.



**Figure 3.5** : Class Diagram for Application and evaluation of a K-Medoids based shape clustering method for an articulated design space

### 3.5 SEQUENCE DIAGRAM

The sequence diagram illustrates the interaction among various components in the clustering system as a user requests shape clustering. It begins with the User initiating a request through the User Interface (UI) to cluster selected shapes. The UI sends the data to the Preprocessing Module, which cleans and normalizes it before forwarding it to the Shape Clustering Module (SC-KM). The SC-KM then requests distance calculations from the Distance Calculation Module, which fetches shape data from the Database, computes pairwise distances, and returns this information. Using the distance data, the SC-KM performs clustering to identify groups of similar shapes. The results are sent to both the Visualization Module and the Evaluation Module, which create graphical representations and assess clustering quality metrics, respectively. Finally, the processed results are sent back to the UI for display to the User, providing a structured approach to shape clustering that emphasizes preprocessing, execution, and visualization.

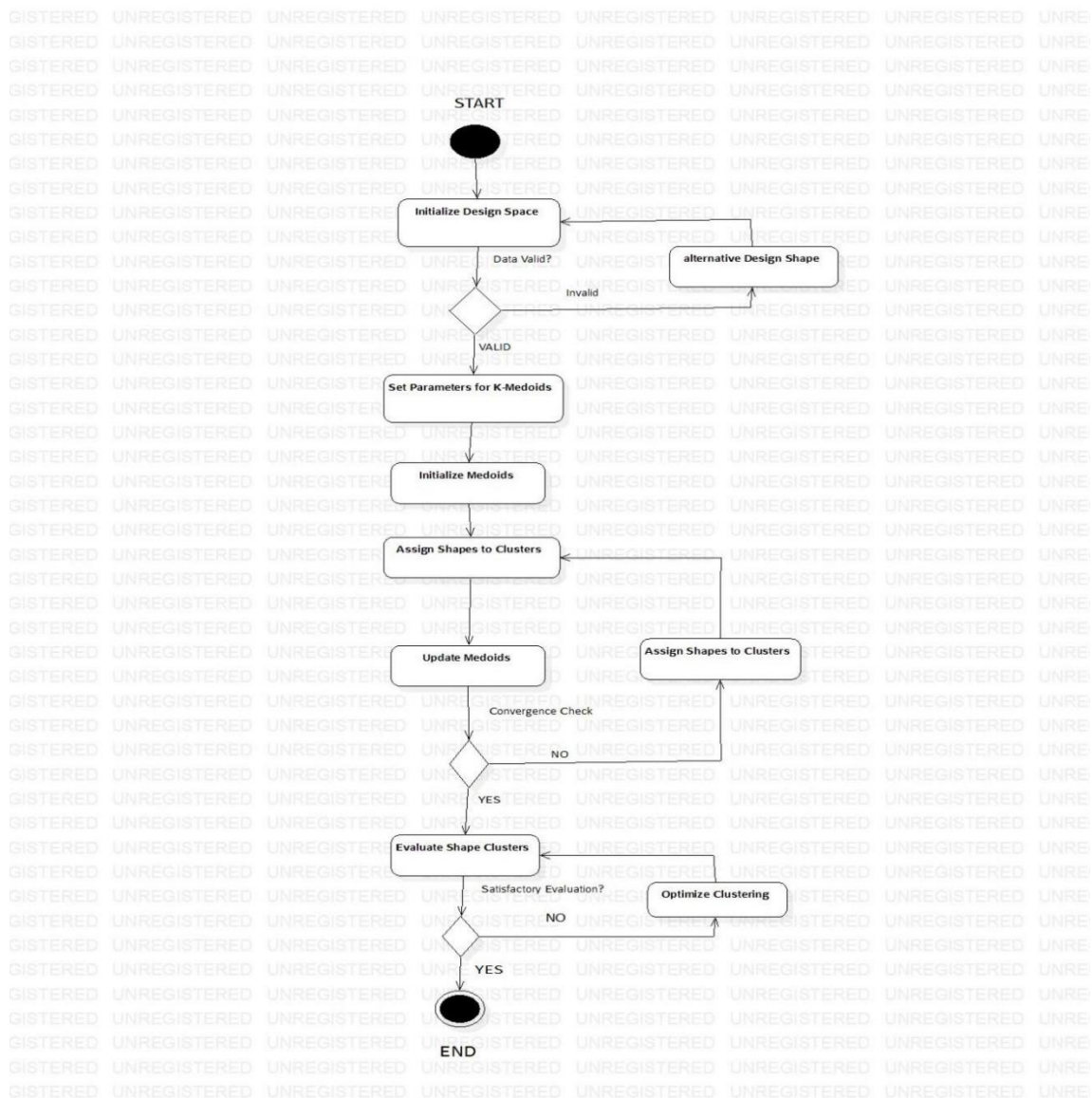


**Figure 3.6 :** Sequence Diagram for Application and evaluation of a K-Medoids based shape clustering method for an articulated design space



### 3.6 ACTIVITY DIAGRAM

The K-Medoids clustering activity diagram starts with the Start point, leading to Initialize Design Space, where shapes are loaded and validated. If valid, the process continues to Set Parameters for K-Medoids otherwise, an error handling path prompts user corrections. After setting parameters, the Initialize Medoids step selects initial medoids and resolves any issues. Assign Shapes to Clusters assigns shapes to the nearest medoid while managing unassignable shapes as outliers. The Update Medoids step recalculates medoids, followed by a convergence check to either iterate or Evaluate Shape Clusters. The evaluation assesses clustering quality; satisfactory results conclude the process, while unsatisfactory ones lead to optimization before potentially re-running the clustering. Finally, the workflow ends.



**Figure 3.7 :** Activity diagram for Application and evaluation of a K-Medoids based shape clustering method for an articulated design space



## **4. IMPLEMENTATION**

## 4. IMPLEMENTATION

### 4.1 SOURCE CODE

```
from tkinter import *

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

from tkinter.filedialog import askopenfilename

import pandas as pd

import numpy as np

import os

import cv2

import numpy as np

from sklearn_extra.cluster import KMedoids

main = tkinter.Tk()

main.title("Application and evaluation of a K-Medoids-based shape clustering method for an articulated design space")

main.geometry("1200x1200")

global dataset

global X, Y, clusters, shape1, shape2, shape3, shape4

def uploadDataset():

    global dataset

    text.delete('1.0', END)

    filename = filedialog.askdirectory(initialdir=".")

    text.insert(END, str(filename) + " Dataset Loaded\n\n")

    pathlabel.config(text=str(filename) + " Dataset Loaded\n\n")

def preprocessDataset():

    text.delete('1.0', END)

    global dataset

    global X, Y
```

```

if os.path.exists("model/X.txt.npy"):
    X = np.load('model/X.txt.npy')
    Y = np.load('model/Y.txt.npy')
else:
    X = []
    Y = []
    for root, dirs, directory in os.walk(dataset):
        for j in range(len(directory)):
            name = os.path.basename(root)
            print(name+" "+root+"/"+directory[j])
            if 'Thumbs.db' not in directory[j]:
                img = cv2.imread(root+"/"+directory[j],0)
                img = cv2.resize(img, (64,64))
                X.append(img.ravel())
                Y.append(getID(name))
    X = np.asarray(X)
    Y = np.asarray(Y)
    np.save('model/X.txt',X)
    np.save('model/Y.txt',Y)
X = X.astype('float32')
X = X/255
test = X[3]
test = test.reshape(64,64)
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
Y = Y[indices]
text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n\n")
text.insert(END,"Shapes found in dataset are : Star, Circle, Triangle and Rectangle\n\n")

```

```
text.update_idletasks()

test = cv2.resize(test,(300,300))

cv2.imshow("Sample Process Image",test)

cv2.waitKey(0)

def runClustering():

    global X, Y, clusters

    global shape1, shape2, shape3, shape4

    text.delete('1.0', END)

    kmedoids = KMedoids(n_clusters=4, metric='euclidean',random_state=0,max_iter=10000)

    kmedoids.fit(X)

    clusters = kmedoids.labels_

    text.insert(END,"Clusters List = "+str(clusters.tolist())+"\n\n")

    count1 = 0

    count2 = 0

    count3 = 0

    count4 = 0

    shape1 = []

    for i in range(len(clusters)):

        if clusters[i] == 0:

            count1 += 1

            if len(shape1) < 20:

                if clusters[i] == 0:

                    shape1.append(X[i])

    shape2 = []

    for i in range(len(clusters)):

        if clusters[i] == 1:

            count2 += 1

            if len(shape2) < 20:

                if clusters[i] == 1:
```

```
        shape2.append(X[i])
shape3 = []
for i in range(len(clusters)):
    if clusters[i] == 2:
        count3 += 1
    if len(shape3) < 20:
        if clusters[i] == 2:
            shape3.append(X[i])
shape4 = []
for i in range(len(clusters)):
    if clusters[i] == 3:
        count4 += 1
    if len(shape4) < 20:
        if clusters[i] == 3:
            shape4.append(X[i])
text.insert(END,"Clustering process completed\n\n")
text.insert(END,"Total shapes found in Cluster1 : "+str(count1)+"\n\n")
text.insert(END,"Total shapes found in Cluster2 : "+str(count2)+"\n\n")
text.insert(END,"Total shapes found in Cluster3 : "+str(count3)+"\n\n")
text.insert(END,"Total shapes found in Cluster4 : "+str(count4)+"\n\n")
def plotShapes(array, titles):
    w = 64
    h = 64
    fig = plt.figure(figsize=(8, 8))
    columns = 4
    rows = 5
    index = 0
    for i in range(1, columns*rows +1):
        img = array[index].reshape(64,64)
```

```
fig.add_subplot(rows, columns, i)

plt.imshow(img)

index += 1

fig.suptitle(titles, fontsize=20)

def visualization():

    global shape1, shape2, shape3, shape4

    plotShapes(shape1, "Shapes in Cluster 1")

    plotShapes(shape2, "Shapes in Cluster 2")

    plotShapes(shape3, "Shapes in Cluster 3")

    plotShapes(shape4, "Shapes in Cluster 4")

    plt.show()

def close():

    main.destroy()

font = ('times', 14, 'bold')

title = Label(main, text='Application and evaluation of a K-Medoids-based shape clustering  
method for an articulated design space')

title.config(bg='DarkGoldenrod1', fg='black')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=5,y=5)

font1 = ('times', 12, 'bold')

uploadButton = Button(main, text="Upload Shapes Dataset", command=uploadDataset)

uploadButton.place(x=50,y=100)

uploadButton.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='brown', fg='white')

pathlabel.config(font=font1)

pathlabel.place(x=560,y=100)

preprocessButton = Button(main, text="Preprocess & Hamming Distance Calculation",  
command=preprocessDataset)
```

```
preprocessButton.place(x=50,y=150)
preprocessButton.config(font=font1)
hybridMLButton = Button(main, text="Run K-Medoids Clustering Algorithm",
command=runClustering)
hybridMLButton.place(x=50,y=200)
hybridMLButton.config(font=font1)
snButton = Button(main, text="Similar Shapes Visualization from Clusters",
command=visualization)
snButton.place(x=50,y=250)
snButton.config(font=font1)
snButton = Button(main, text="Exit", command=close)
snButton.place(x=50,y=300)
snButton.config(font=font1)
font1 = ('times', 12, 'bold')
text=Text(main,height=25,width=100)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=400,y=150)
text.config(font=font1)
main.config(bg='LightSteelBlue1')
main.mainloop()
```

```
#TEST TRAIN
```

```
import os
```

```
import cv2
```

```
import numpy as np
```

```
from sklearn_extra.cluster import KMedoids
```

```
import matplotlib.pyplot as plt
```

```
# Non-Binary Image Classification using Convolution Neural Networks
```

```
'''
```

```
path = 'shapes'
labels = []
X_train = []
Y_train = []
def getID(name):
    index = 0
    for i in range(len(labels)):
        if labels[i] == name:
            index = i
            break
    return index
for root, dirs, directory in os.walk(path):
    for j in range(len(directory)):
        name = os.path.basename(root)
        if name not in labels:
            labels.append(name)
print(labels)
for root, dirs, directory in os.walk(path):
    for j in range(len(directory)):
        name = os.path.basename(root)
        print(name+" "+root+"/"+directory[j])
        if 'Thumbs.db' not in directory[j]:
            img = cv2.imread(root+"/"+directory[j],0)
            img = cv2.resize(img, (64,64))
            X_train.append(img.ravel())
            Y_train.append(getID(name))

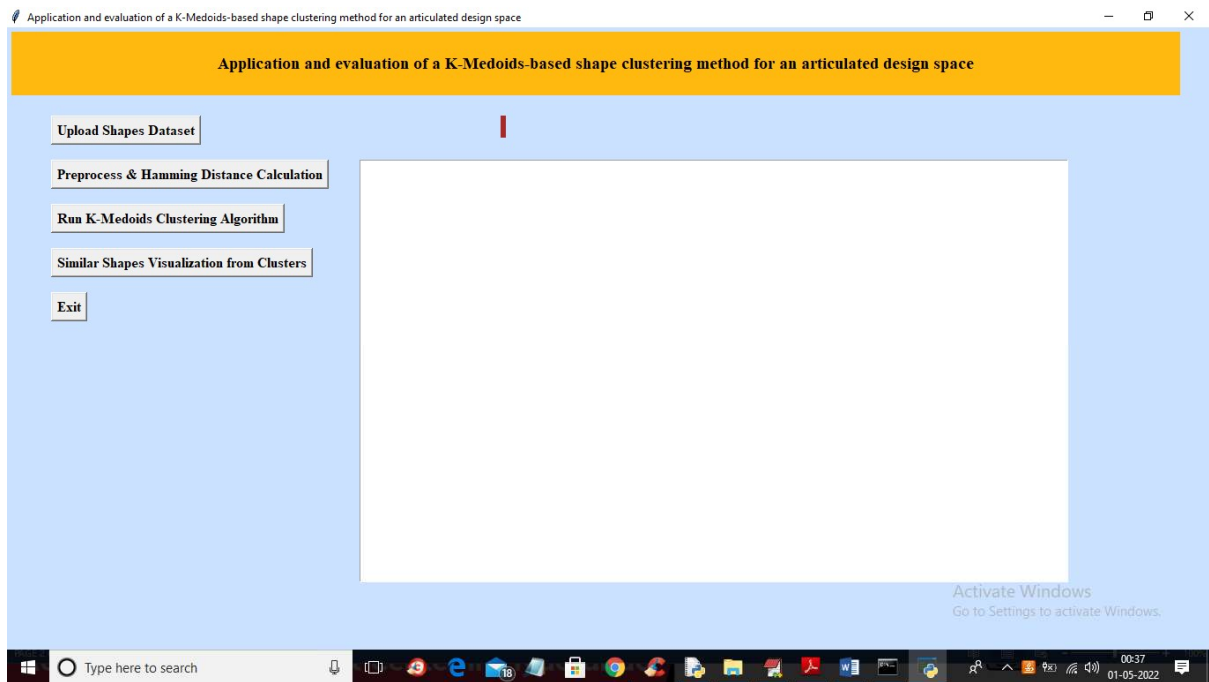
X_train = np.asarray(X_train)
Y_train = np.asarray(Y_train)
```



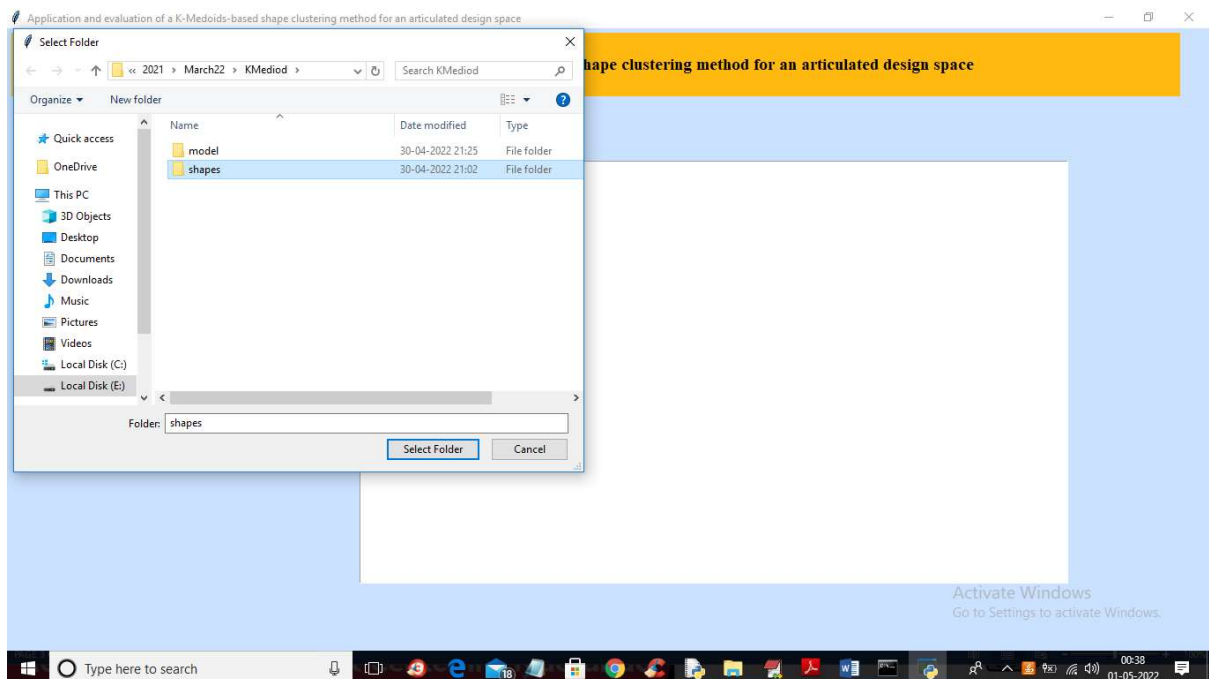
```
print(Y_train)
np.save('model/X.txt',X_train)
np.save('model/Y.txt',Y_train)
'''
X_train = np.load('model/X.txt.npy')
Y_train = np.load('model/Y.txt.npy')
X_train = X_train.astype('float32')
X_train = X_train/255
test = X_train[3]
test = test.reshape(64,64)
cv2.imshow("aa",test)
cv2.waitKey(0)
indices = np.arange(X_train.shape[0])
np.random.shuffle(indices)
X_train = X_train[indices]
Y_train = Y_train[indices]
kmedoids = KMedoids(n_clusters=4, metric='euclidean',random_state=0,max_iter=10000)
kmedoids.fit(X_train)
predict = kmedoids.labels_
images = []
for i in range(len(predict)):
    if len(images) < 20:
        if predict[i] == 0:
            images.append(X_train[i])
images1 = []
for i in range(len(predict)):
    if len(images1) < 20:
        if predict[i] == 1:
            images1.append(X_train[i])
```

```
images2 = []
for i in range(len(predict)):
    if len(images2) < 20:
        if predict[i] == 2:
            images2.append(X_train[i])
images3 = []
for i in range(len(predict)):
    if len(images3) < 20:
        if predict[i] == 3:
            images3.append(X_train[i])
def plotShapes(array, title):
    w = 64
    h = 64
    fig = plt.figure(figsize=(8, 8))
    columns = 4
    rows = 5
    index = 0
    for i in range(1, columns*rows + 1):
        img = array[index].reshape(64,64)
        fig.add_subplot(rows, columns, i)
        plt.imshow(img)
        index += 1
    plt.title(title)
    plt.show()
plotShapes(images,"Shapes in Cluster 1")
plotShapes(images1,"Shapes in Cluster 2")
plotShapes(images2,"Shapes in Cluster 3")
plotShapes(images3,"Shapes in Cluster 4")
```

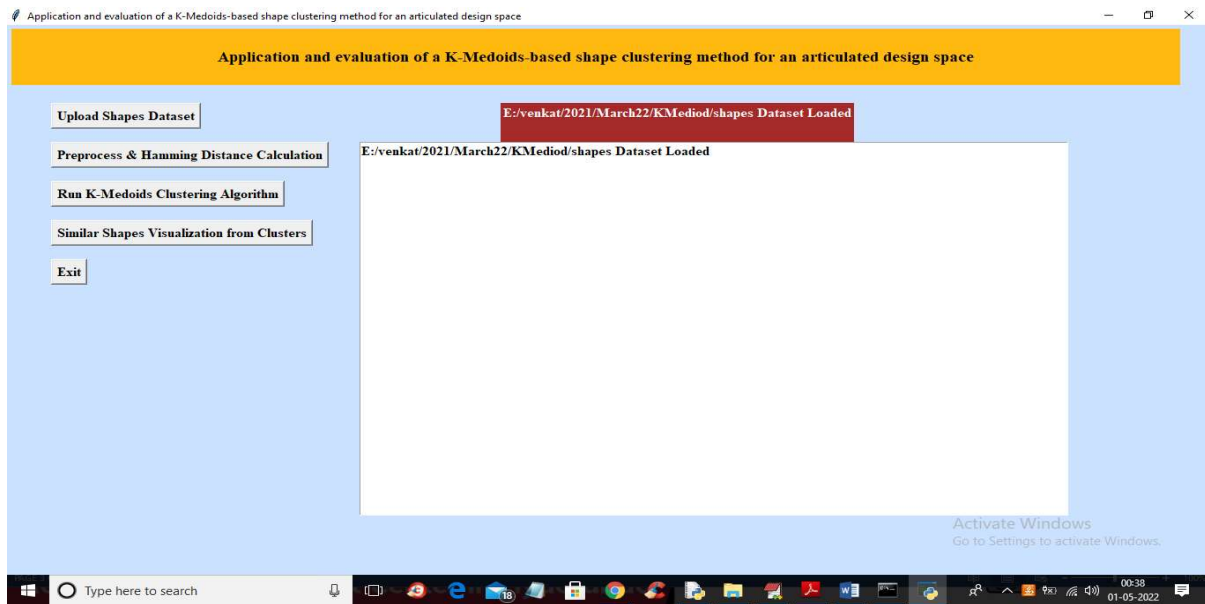
## **5. RESULTS**



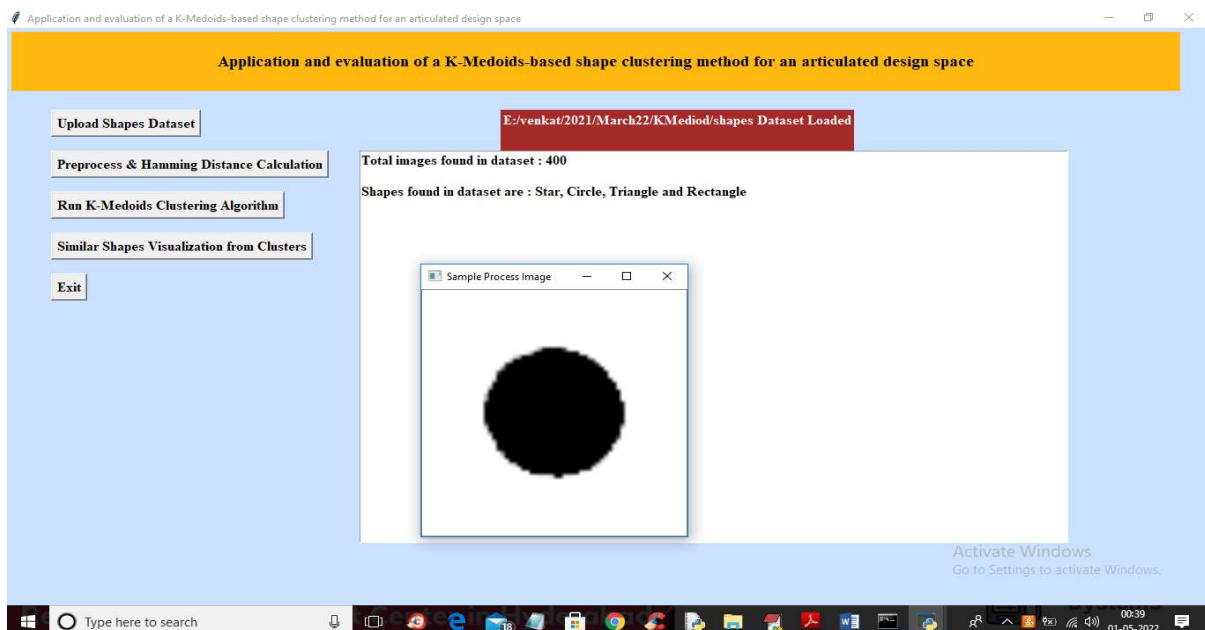
**RESULT 5.1 :** In above screen click on ‘Upload Shapes Dataset’ button to upload shapes to application



**RESULT 5.2 :** In above screen selecting and uploading ‘shapes’ folder and then click on ‘Select Folder’ to load dataset and get below output.

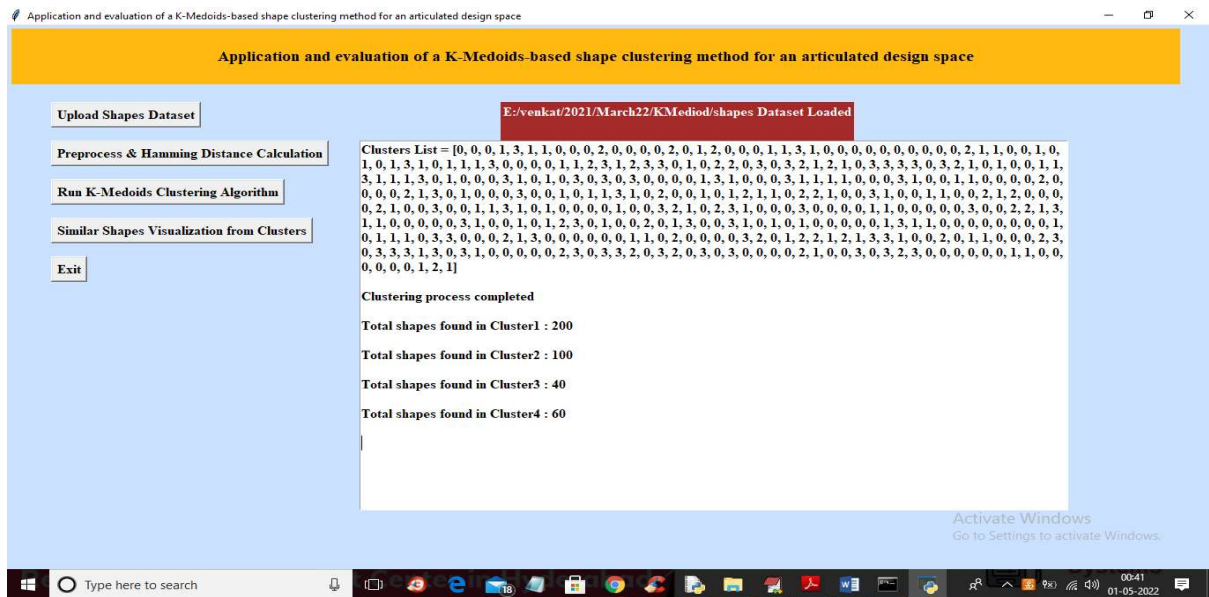


**RESULT 5.3 :** In above screen dataset loaded and now click on 'Preprocess & Hamming Distance Calculation' button to read all shapes and then compute hamming distance



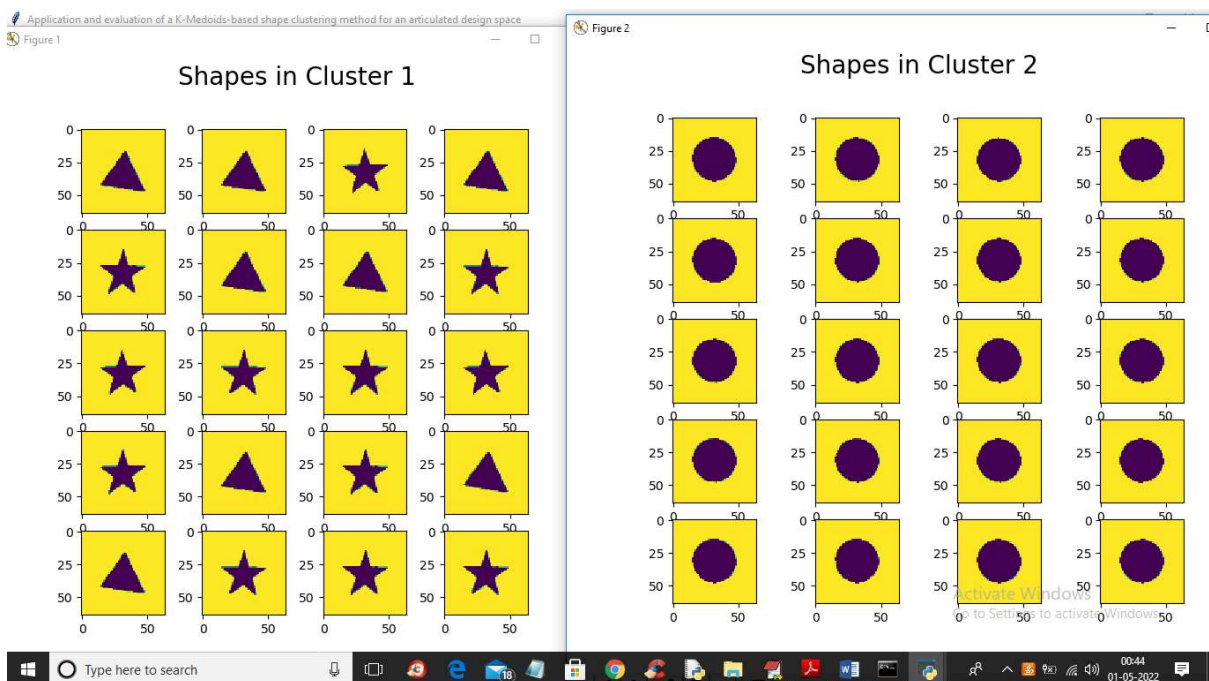
**RESULT 5.4 :** In above screen all shapes are processed and saved in application memory

With use of hamming distance and we can see dataset contains 400 shapes of different shapes types and to check weather shapes are processed properly or not so I am displaying one sample processed image and now close above image and then click on 'Run K-Medoids Clustering Algorithm' button to cluster all shapes and get below output

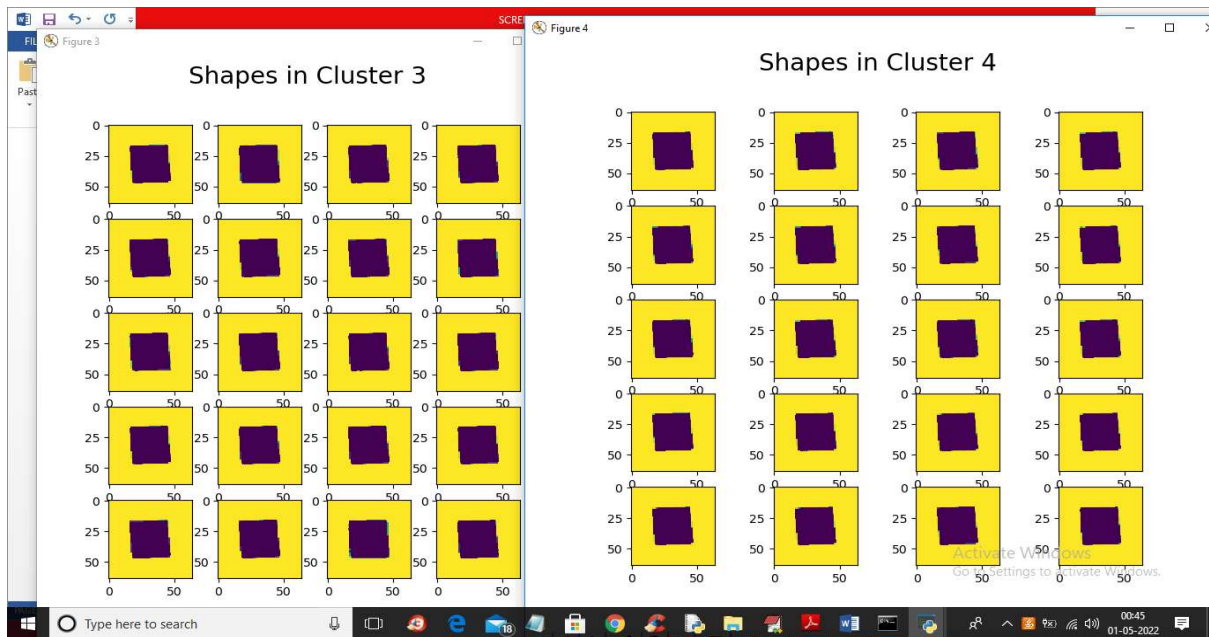


**RESULT 5.5 :** In above screen we can see clustering process completed and in cluster list

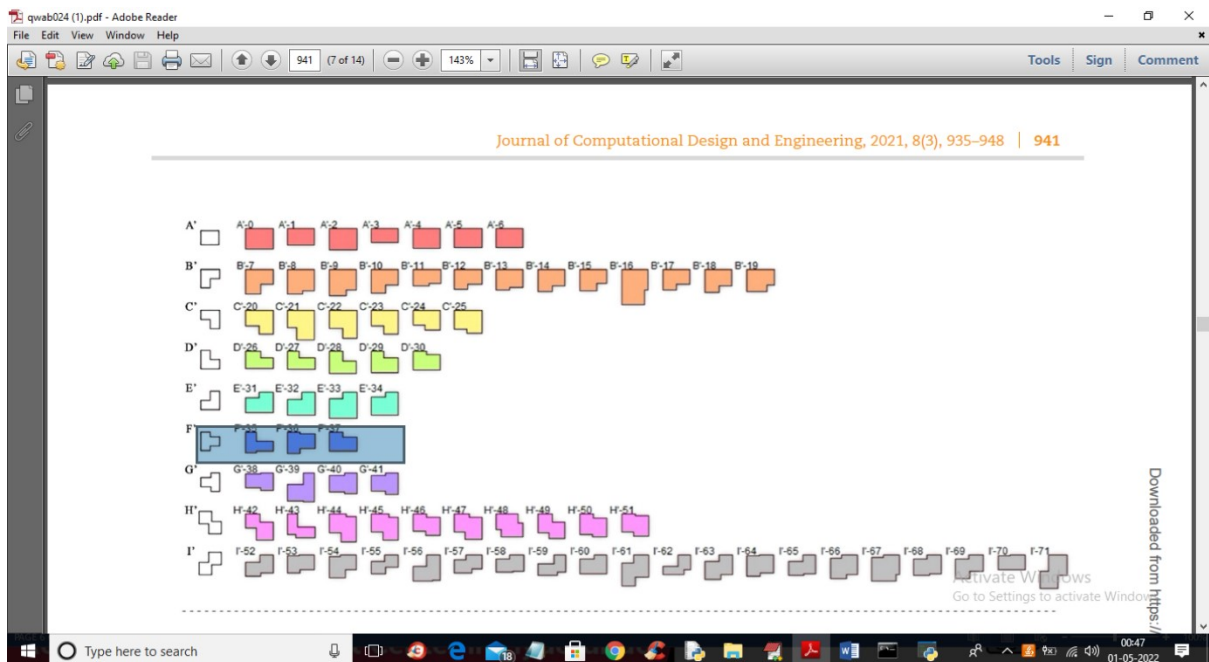
we can see which record is went to which cluster for example shape or record 1 goes to cluster and we can see all shapes are arranged between cluster 0 to 3 and then we can see how many shapes are arranged in each cluster and now click on ‘Similar Shapes Visualization from Clusters’ button to visualize shapes in different clusters



**RESULT 5.6 :** In above screen we can see cluster 1 contains all triangles and stars shapes (here triangle and start is having little similarity in shapes) and cluster 2 contains all circles shapes and below is 3<sup>rd</sup> and 4<sup>th</sup> cluster shapes output



**RESULT 5.7 :** In above screen clusters 3 and 4 contains similar shape called Rectangle and in below screen from paper we can see different shapes exists in same cluster



**RESULT 5.8 :** In above screen from paper we can see cluster F contains different shapes which look little similar

**The main aim of the project is to group same shapes in to same cluster**

## **6. TESTING**



## **6.1 INTRODUCTION OF TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## **6.2 TYPES OF TESTING**

### **6.2.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **6.2.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## 6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## 6.3 TEST CASES

### 6.3.1 UPLOADING DATASET

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads Model data (X.txt.npy)	It is Used as sample data set To cluster	The user uploads the models datasets Of X.txt.npy	Uploaded Successfully
2	User uploads Shape data	It is Used as sample data set to cluster	The User uploads the various shapes datasets	Uploaded Successfully
3	User uploads Model data (Y.txt.npy)	It is Used as sample data set To cluster	The user uploads the models datasets Of Y.txt.npy	Uploaded Successfully

### 6.3.2 PROCESSING DATASET

Test case ID	Test case name	Purpose	Input	Output
1	Process 1	Preprocess & Hamming Distance Calculation	The shape dataset is given	Shapes will be visualized
2	Process 2	Run K-Medoids Clustering Algorithm	The shape or models dataset is given	Clusters List will be shown
3	Process 3	Similar Shapes Visualization from Clusters	The shape or models dataset is given	Various shape will be Clustered from datasets
4	Process 4	Upload data set	The Models and Shapes Dataset is given	The Dataset will be Uploaded

# **7. CONCLUSION & FUTURE SCOPE**

## **7. CONCLUSION AND FUTURE SCOPE**

### **7.1 PROJECT CONCLUSION**

The "Application and Evaluation of a K-Medoids Based Shape Clustering Method for an Articulated Design Space" project has successfully fulfilled its objectives, presenting an innovative clustering approach specifically tailored for architectural designs. The Shape Clustering using K-Medoids (SC-KM) method has been implemented effectively, with a focus on organizing and simplifying the design space in a computationally efficient manner.

The project was developed with a clear modular structure, ensuring that each component operates independently yet seamlessly integrates into the overall system. All aspects of the system were thoroughly tested, using both ideal and challenging data sets, and the SC-KM method demonstrated its robustness and precision in handling a variety of design shapes.

Through the introduction of an algorithm to convert boundary-based shapes into grid-based shapes, the system has shown significant flexibility in accommodating different design representations. The clustering evaluation metrics used to assess the system's performance have confirmed its effectiveness in improving the design process by reducing complexity and computational requirements.

The system's architecture is designed for future scalability, making it adaptable to more complex design spaces or other related applications, such as image processing. The project aligns with the goals outlined in the design phase, and the results validate the viability of the SC-KM method as a useful tool for architects and designers.

In conclusion, the project provides a practical solution for clustering shapes within the architectural domain, offering insights into the development of advanced design tools. With the potential for future enhancements, this work stands as a significant contribution to the field of generative design systems and computational design.

## 7.2 FUTURE SCOPE

The "Application and Evaluation of a K-Medoids Based Shape Clustering Method for an Articulated Design Space" has laid a strong foundation for future development. Additional clustering techniques, such as hierarchical or density-based methods, can be integrated to expand the system's capabilities. Moreover, incorporating deep learning-based approaches for automated feature extraction from architectural designs can enhance clustering accuracy.

The system can also be extended to handle more complex design spaces, including 3D models or parametric designs, allowing for broader applications. The existing architecture supports scalability, making it adaptable to larger datasets or more intricate clustering tasks. Integrating cloud computing resources could further improve the system's efficiency and processing power for handling large-scale architectural projects.

Additionally, the method can be applied to other fields such as image processing or urban planning, providing a versatile tool for different domains that require shape clustering or generative design systems. The system can also support integration with various databases and design software for more seamless data management and collaboration in the future.

## **8. BIBLIOGRAPHY**

## 8. BIBLIOGRAPHY

### 8.1 REFERENCE

1. Martínez-Rocamora, Alejandro, Pilar Díaz-Cuevas, Juan Camarillo-Naranjo, David Gálvez-Ruiz, and Patricia González-Vallejo. "Identification of residential building typologies by applying clustering techniques to cadastral data." *Journal of Building Engineering* 86 (2024): 108912.
2. Heidari, J., N. Daneshpour, and A. Zangeneh. "A novel K-means and K-medoids algorithms for clustering non-spherical-shape clusters non-sensitive to outliers." *Pattern Recognition* (2024): 110639.
3. Zahrotun, Lisna, Utaminingsih Linarti, Banu Harli Trimulya Suandi As, Herri Kurnia, and Liya Yusrina Sabila. "Comparison of K-Medoids Method and Analytical Hierarchy Clustering on Students' Data Grouping." *JOIV: International Journal on Informatics Visualization* 7, no. 2 (2023): 446-454.
4. Krishna, Raji, R. Suganya, Mounika Sai Yaganti, and Seyed Mohamed Buhari. "Shape Organization for Enhanced Design Optimization in Generative Systems Using Clustering Technology." In *2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC)*, pp. 569-574. IEEE, 2023..
5. Goparaju, Anupama, Krithika Iyer, Alexandre Bône, Nan Hu, Heath B. Henninger, Andrew E. Anderson, Stanley Durrleman et al. "Benchmarking off-the-shelf statistical shape modeling tools in clinical applications." *Medical image analysis* 76 (2022): 102271.
6. Grove, Jared Thomas. "A k-medoids approach to exploring districting plans." PhD diss., The University of Iowa, 2022.
7. Yousif, Shermeen, and Wei Yan. "Application and evaluation of a K-Medoids-based shape clustering method for an articulated design space." *Journal of Computational Design and Engineering* 8, no. 3 (2021): 935-948.



8. SREE, K. MEGHA. "VULNERABILITY ANALYSIS USING UNSUPERVISED MACHINE LEARNING METHODS."
9. Wang, X., and X. Wang. "A fast k-medoids clustering algorithm for image segmentation based object recognition." *J. Rob. Autom* 4 (2021): 202-211.
10. Gao, Yunkai, Yuexing Duan, James Yang, Zhe Liu, and Chao Ma. "Improved K-medoids algorithm-based clustering analysis for handle driving force in automotive manual sliding door closing process." *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 235, no. 2-3 (2021): 871-880.

## 8.2 WEBSITES

1. <https://doi.org/10.5281/zenodo.888148>
2. <https://doi.org/10.4135/9781412986397>
3. <https://doi.org/10.1260/0266351991494722>
4. . <https://doi.org/10.1017/S0890060418000033>

## PROJECT GITHUB LINK

<https://github.com/Raghuvaran0557/application-evaluation-of-a-K--Medoids-based-shape-clustering>