# SCULPTURE PREDICTION

A Course Project report submitted

in partial fulfilment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

in

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

by


| | |
|---|---|
| **T.SAI TEJA** | **(2103A52035)** |
| **K. MITHRA SRI** | **(2103A52050)** |
| **N. ANUHYA** | **(2103A52186)** |


Under the guidance of

**Mr. Ramesh**

Assistant Professor, Department of CSE.



**Department of Computer Science and Artificial Intelligence**

**Department of Computer Science and Artificial Intelligence**

## CERTIFICATE

This is to certify that project entitled **"SCULPTURE PREDICTION**" is the bonafied work carried out by **Sai Teja, Mithrasri, Anuhya** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.


**Mr. Ramesh**
Asst. Professor,
(CSE),
S R University,
Ananthasagar, Warangal.

**Dr. M.Sheshikala**
Assoc. Prof. & HOD

S R University,
Ananthasagar, Warangal.

# ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. Ramesh, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M.Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

# ABSTRACT

This project is to predict the sculpture and provide the information about that particular sculpture. This project contains 2 types of sculpture from two temples.The dataset is about the pictures of thousand pillar nandi and Warangal fort nandi. This project is helpful for the user to know whether the nandhi sculpture belongs to either of the place. This dataset has 283 images from warangal fort and 264 images from thousand pillar temple. The sculpture detection project aims to develop a computer vision system capable of detecting and recognizing sculptures in digital images. The project will involve training a deep learning model on a large dataset of sculpture images, using techniques such as convolutional neural networks (CNNs) and transfer learning. The resulting model will be able to identify various types of sculptures, including abstract, figurative, and architectural sculptures, as well as different materials and styles.

The detection and recognition of sculptures in images can have various practical applications, including art history research, cultural heritage preservation, and museum collections management. It can also serve as a tool for promoting public engagement with art and cultural heritage by allowing users to search and explore digital collections of sculptures.

# Table of Contents

# 1.INTRODUCTION

## 1.1. Overview

Sculpture detection refers to the use of computer vision techniques to identify and analyze sculptures in images or videos. It involves the development of algorithms that can automatically detect the presence of sculptures in an image, distinguish them from other objects in the image, and extract relevant features such as the sculpture's shape, texture, and color. Sculpture detection has a wide range of applications, including art authentication, museum and gallery curation, and cultural heritage preservation. By automating the process of identifying and analyzing sculptures, it can help art experts and researchers to more efficiently and accurately study and preserve these important cultural artifacts.

Machine learning techniques involve the use of algorithms and statistical models that enable computers to learn from data and make predictions or decisions based on that data. Some common machine learning techniques used in sculpture detection include:

Supervised learning: Supervised learning algorithms are trained on labeled data, where each image is labeled as either containing a sculpture or not. The algorithm then learns to classify new images based on the features it has learned from the training data.

Unsupervised learning: Unsupervised learning algorithms are used to identify patterns and relationships in unlabeled data. In sculpture detection, unsupervised learning can be used to identify common features or characteristics of sculptures across different images.

Deep learning: Deep learning is a type of machine learning that uses neural networks to learn and make predictions based on large amounts of data. In sculpture detection, deep learning can be used to automatically extract features from images and classify them based on those features.

## 1.2. Problem Statement:

Before the development of sculpture detection technology, people who needed to identify and analyze sculptures in images or videos faced several challenges. Here are some of the problems they might have faced:

1. Time-consuming and labor-intensive process: Identifying and analyzing sculptures in images or videos was a manual process that required extensive knowledge and expertise in art history, sculpture classification, and visual analysis. This process could take a significant amount of time and could be labor-intensive.

2. Subjectivity and errors: Without the aid of technology, identifying and analyzing sculptures was subject to human error and bias. Experts would have to carefully examine each image or video frame to identify the presence of sculptures and manually extract features such as shape, texture, and color. This process could be subject to errors or inconsistencies in interpretation.

3. Limited ability to analyze large collections: Without the aid of technology, it was difficult to analyze large collections of sculptures or to compare and contrast sculptures across different locations or time periods. This limited the ability of art experts to study and preserve these important cultural artifacts and hindered research in the field of art history.

4. Limited access to sculptures: Some sculptures might be located in remote or inaccessible locations, making it difficult for experts to study and analyze them. Without the aid of technology, it was difficult to remotely analyze sculptures or to share information about them across different locations.

Overall, the lack of sculpture detection technology made the process of identifying and analyzing sculptures more difficult and time-consuming, and limited the ability of art experts to study and preserve these important cultural artifacts. The development of sculpture detection technology has addressed many of these challenges and has revolutionized the field of art history and cultural heritage preservation.

## 1.3. Existing solution

The existing systems for sculpture detection can be broadly categorized into two types: traditional computer vision techniques and deep learning-based approaches.

Traditional computer vision techniques include methods such as feature extraction, object recognition, and classification. These techniques rely on hand-crafted features, such as color, texture, and shape, to identify and classify sculptures in images or videos. They often require domain-specific knowledge and can be computationally expensive. Examples of traditional computer vision techniques for sculpture detection include SIFT (Scale-Invariant Feature Transform) and HOG (Histogram of Oriented Gradients).

Deep learning-based approaches, on the other hand, use neural networks to automatically learn features from images or videos. These methods have shown great success in a wide range of computer vision tasks, including sculpture detection. Deep learning-based approaches often require large amounts of labeled data and can be computationally intensive during training, but they can achieve state-of-the-art results. Examples of deep learning-based approaches for sculpture detection include Faster R-CNN (Region-based Convolutional Neural Network) and YOLO (You Only Look Once).

There are also some existing systems that combine traditional computer vision techniques with deep learning-based approaches to improve the accuracy of sculpture detection. These hybrid systems can leverage the strengths of both approaches to achieve better performance.

## 1.4. PROPOSED SYSTEM

A proposed system for sculpture detection would leverage deep learning-based approaches to automatically identify and analyze sculptures in images or videos. The proposed system could include the following components:

- **Data collection**: The system would need a large dataset of images or videos of sculptures to train the deep learning models. This data could be collected from various sources, such as museum collections, public art installations, and online image repositories.

- **Preprocessing**: The images or videos in the dataset would need to be preprocessed to extract features such as color, texture, and shape. This could involve techniques such as image segmentation and feature extraction.

- **Training**: The system would use deep learning models, such as Convolutional Neural Networks (CNNs), to learn features from the preprocessed images or videos. The models would be trained on a labeled dataset, where each sculpture is labeled with its corresponding class.

- **Testing and evaluation**: Once the models are trained, they would be tested on a separate set of images or videos to evaluate their performance. The performance could be evaluated using metrics such as precision, recall, and F1 score.

- **Deployment**: The trained models could be deployed in a software application that allows users to upload images or videos and automatically detect and classify sculptures. The application could also provide additional information about each sculpture, such as its history, artist, and location.

Overall, a proposed system for sculpture detection would leverage deep learning models to automate the process of identifying and analyzing sculptures in images or videos. This would significantly reduce the time and labor required for manual analysis, and enable art experts to study and preserve sculptures more efficiently and accurately.

# 2.LITERATURE SURVEY

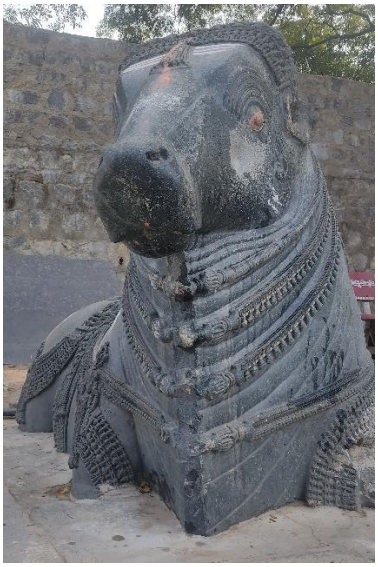
Sculpture dectection is an idea not been given much attention yet. It is a noval one, where much work has Not been devoted to it. In this field, a lot of research has not yet been made. There have been many different approaches and strategies shown and implemented by various researchers and authors. The section below shows the given out findings of the key and important aspects each used by the authors in various Research papers.A member from IACSIT worked on a sculpture detection model which he called the Thai Buddhist Sculpture Recognition System[1]. The main target of this system was the recognition of monuments relating to Buddhism in Thailand. This system consists of five parts viz. Image acquisition (via a camera), image pre-processing, Feature Ex-traction, Image recognition(applied using the Euclidean Distance technique), and dis-playing the output through a graphic user interface.Another team worked on the similar concepts, i.e, Sculpture Recognition using a different algorithm[2]. They combined and compared various algorithms which gave out the same result to check the most efficient one.The SIFT algorithm was paired with K-Nearest Neighbours, Support Vector Machine, and Artificial Neural Network using four different approaches –Min, max,mean, and median key paddings. CNN wax`s used as well. They used Contrast Limited Adaptive Histogram Equalization to improve the efficiency of the overall system. The most efficient and accurate model they worked out was that of CNN with the incorporationof CLAHE. A total of around 2500 images (224*224 pixels) from 15 different entity classes(Indian deities) were used training and 20 for testing.A group of six people from IIT Roorkee attempted to tackle a similar problem. They made a model that recognized monuments in India using Deep Convolutional Neural Networks[3]. Several images of Indian monuments from various angles were fed into the model. They acquired a very high amount of accuracy. Feature extraction was car-ried out using the Histogram of oriented gradients. Training of the data was done using Local binary patterns and GIST features.

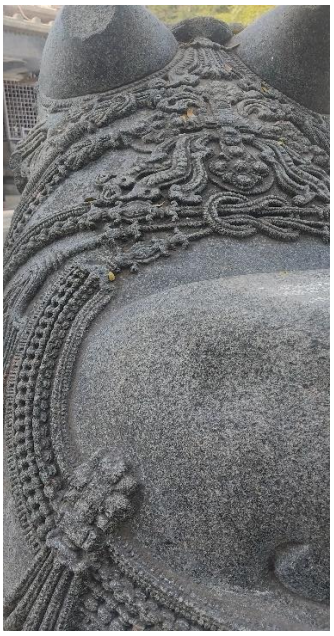
# 3.DATA PREPROCESSING

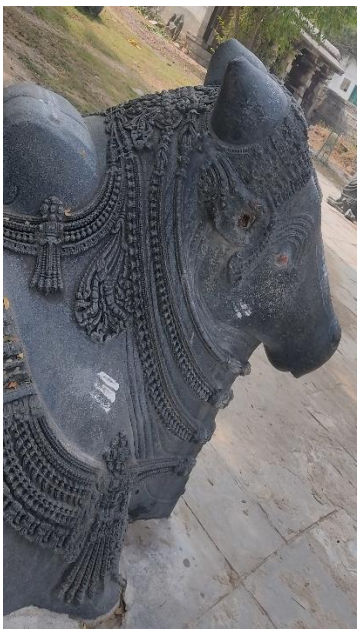
## 3.1. Dataset Description

- This dataset contains nandi pictures.
- Here we have taken nandi pictures from two temples (Warangal fort and thousand pillar temple)
- The entire dataset has the pictures but no statistical data provided.

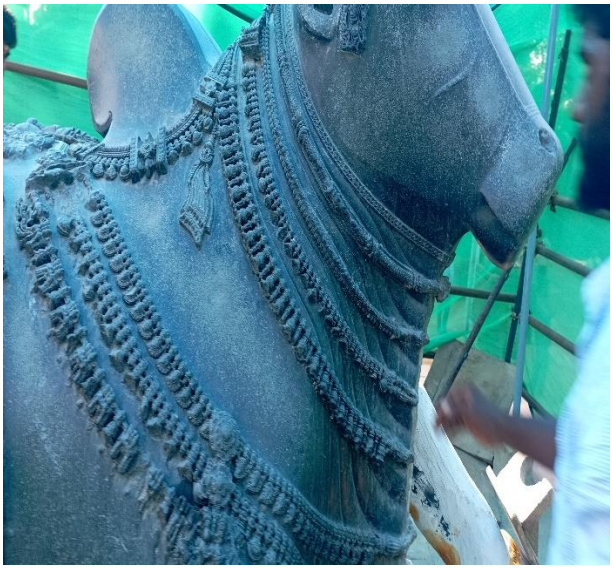
**Some pictures of Warangal fort nandi:**





- **Some pictures of Thousand pillar temple nandi:**



### 3.2. Data Cleaning

Data cleaning is an important step in any machine learning task, including car object detection. It involves identifying and correcting errors and inconsistencies in the dataset to ensure that the data is accurate, complete, and ready for analysis.

Here are some common data cleaning steps that can be applied to sculpture detection datasets:

Remove duplicates: Duplicates can occur when multiple images of the same scene are captured. Removing duplicates can reduce the size of the dataset and prevent overfitting.

Remove outliers: Outliers can occur when the dataset contains images that do not represent the typical distribution of the data. Removing outliers can improve the accuracy of the model and prevent it from being biased towards non-representative data.

Check for missing data: Missing data can occur when some of the images in the dataset do not contain the relevant information. Checking for missing data and either removing or imputing missing values can improve the accuracy of the model.

Normalizing:

```
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
d= sc_x.fit_transform(d)#normalizing
e = sc_x.transform(e)
```

Normalize the data: Normalizing the data involves scaling the pixel values of the images to a standard range, which can improve the performance of the model and reduce the impact of lighting and color variations.

Overall, data cleaning is an important step in preparing the dataset for sculpture detection. By removing errors and inconsistencies, and ensuring that the data is accurate and complete, the model can be trained on high-quality data that will lead to more accurate and reliable predictions.

## 3.3. Data Visualization

Here we have taken the images of both Warangal and thousand pillar nandhi pictures.

Now we are going to predict the the nandhi whether it belongs to thousand pillars or waranagal fort.

Here we assigned index values to the 2 folders we took thousand pillar temple as 0 and Warangal fort as 1.

We took thousand pillar nandhi pictures as 0 and Warangalfort nandhi pictures as 1.

This project is used for the tourist when they visited to historical places.

This project helps in providing the information when the tourist scans the sculpture when he wants to know about that particular sculpture.

This is worked on the concept of deep learning is a type of machine learning that uses neural networks to learn and make predictions based on large amounts of data. In sculpture detection, deep learning can be used to automatically extract features from images and classify them based on those features.

# 4.METHODOLOGY

## 4.1. Procedure to solve the given problem

**LOGISTIC REGRESSION**:

- Logistic regression is used to obtain odds ratio in the presence of more than one explanatory variable.
- The procedure is quite similar to multiple linear regression, with the exception that the response variable is binomial.
- The result is the impact of each variable on the odds ratio of the observed event of interest.
- Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.
- A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
- There are three main types of logistic regression: binary, multinomial and ordinal.
- Its features are sepal length, sepal width, petal length, petal width.
- Besides, its target classes are setosa, versicolor and virginica.
- However, it has 3 classes in the target and this causes to build 3 different binary classification models with logistic regression.

**KNN (K-Nearest neighbours):**

- K-Nearest Neighbor (KNN) Algorithm for Machine Learning
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready

**CROSS VALIDATION:**

Cross-validation is a technique used to evaluate the performance of a machine learning model. In k-nearest neighbors (k-NN) algorithm, cross-validation can be used to determine the optimal value of k, which is the number of nearest neighbors to consider for making a prediction.

In k-fold cross-validation, the data is divided into k equal-sized subsets. The algorithm is trained on k-1 of these subsets and tested on the remaining subset. This process is repeated k times, with each subset serving as the testing data once. The results of the k iterations are then averaged to produce a single performance metric.

To use cross-validation in k-NN, you would first split your data into training and testing sets. Then, you would use k-fold cross-validation on the training data to determine the optimal value of k. For each iteration of the cross-validation, you would train the k-NN model on the training data, with a different value of k, and then evaluate its performance on the validation set. After running all k iterations, you would select the value of k that resulted in the best performance, and use that value to train a final model on the entire training set. Finally, you would evaluate the performance of the final model on the testing set.

It is important to note that cross-validation can be computationally expensive, especially for large datasets or high values of k. However, it is a powerful technique for selecting the optimal value of k and ensuring that your model is not overfitting to the training data.

**DECISION TREE:**

- The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.
- A decision tree typically starts with a single node, which branches into possible outcomes.
- Each of those outcomes leads to additional nodes, which branch off into other possibilities.
- This gives it a treelike shape.
- There are three different types of nodes: chance nodes, decision nodes, and end nodes.
- The main components of a decision tree include a root node, decision nodes, chance nodes, alternative branches, and an endpoint node.
- Optional features include rejected alternatives.

**RANDOM FOREST:**

- Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result.
- Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.
- The random forest is a classification algorithm consisting of many decisions trees.
- It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.
- It is a set of Decision Trees.
- Each Decision Tree is a set of internal nodes and leaves.
- In the internal node, the selected feature is used to make decision how to divide the data set into two separate sets with similar responses within.

**SVM (Support Vector Machine):**

- Support Vector Machine Algorithm
- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

**NAVIE BAYES:**

Naïve Bayes is a simple probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on Bayes' theorem, which describes the probability of an event given prior knowledge of related events. In Naïve Bayes, the input variables are assumed to be independent of each other, which simplifies the computation of the probability of the output variable given the input variables.

The algorithm works by first training a model on a labeled dataset, which involves calculating the probabilities of each input variable given each class label. These probabilities are then used to calculate the probability of each class label given a set of input variables using Bayes' theorem. The class label with the highest probability is then assigned to the input variables.

Naïve Bayes is particularly useful for text classification tasks, such as spam filtering, sentiment analysis, and topic classification. It can also be used for other classification tasks, such as image classification, as long as the assumption of input variable independence holds.

One of the advantages of Naïve Bayes is that it is relatively fast and requires less training data compared to other more complex machine learning algorithms. However, its performance may be limited by the assumption of input variable independence, which may not hold in some real-world applications.

**Confussion Matrix:**

A confusion matrix is a performance evaluation tool used in machine learning and classification tasks. It is a table that summarizes the performance of a classification model by comparing the predicted labels with the actual labels of a set of data.

The confusion matrix typically consists of four values: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

- True positives (TP) represent the number of correct positive predictions made by the model.

- True negatives (TN) represent the number of correct negative predictions made by the model.

- False positives (FP) represent the number of incorrect positive predictions made by the model.

- False negatives (FN) represent the number of incorrect negative predictions made by the model.

These values can be used to calculate various performance metrics such as accuracy, precision, recall, and F1 score.

Accuracy is the proportion of correctly classified data samples out of the total number of samples. Precision measures the proportion of true positive predictions out of all positive predictions made by the model. Recall measures the proportion of true positive predictions out of all actual positive samples. F1 score is a harmonic mean of precision and recall.

The confusion matrix is a useful tool to assess the performance of a classification model and can help identify where the model is making errors. It can be especially useful when the classes are imbalanced or when the cost of false positives and false negatives is different.

Representation:

[[TP FN]

[FP TN]]

**User Modules:**

matplotlib.pyplot

sklearn.model_selection

sklearn.neighbors

sklearn.naive_bayes

sklearn.metrics

sklearn

sklearn.tree

sklearn.ensemble

os

numpy

- matplotlib.pyplot: matplotlib.pyplot is a Python library used for creating visualizations like graphs, charts, and plots. It is a part of the matplotlib library, which is one of the most widely used data visualization libraries in Python.

- sklearn.model_selection: sklearn.model_selection is a module in the scikit-learn library that provides tools for splitting datasets into training and testing sets, cross-validation, and hyperparameter tuning.

- sklearn.neighbors: sklearn.neighbors is a module in the scikit-learn library that provides tools for implementing various types of nearest neighbor algorithms, including K-Nearest Neighbors (KNN).

- sklearn.naive_bayes: sklearn.naive_bayes is a module in the scikit-learn library that provides tools for implementing Naive Bayes classifiers, a family of probabilistic classifiers based on Bayes' theorem.

- sklearn.metrics: sklearn.metrics is a module in the scikit-learn library that provides tools for evaluating the performance of machine learning models. It includes a wide variety of metrics for classification, regression, and clustering tasks.

- Sklearn: scikit-learn, also known as sklearn, is a popular open-source library for machine learning in Python. It provides a wide range of tools for various tasks in machine learning, including data preprocessing, feature selection, model selection, and evaluation.
  the examples for sklearn
  1. Loading and preprocessing datasets
  2. Splitting data into training and testing sets
  3. Creating and training machine learning models, such as decision trees, logistic regression, K-Nearest Neighbors, and Naive Bayes classifiers
  4. Tuning hyperparameters of machine learning models using grid search or random search
  5. Evaluating the performance of machine learning models using various metrics such as accuracy, precision, recall, and F1-score
  6. Visualizing data and results using various plotting functions and tools

- sklearn.tree: sklearn.tree is a module in the scikit-learn library that provides tools for implementing decision tree-based models, including decision trees, random forests, and gradient boosting trees.

- sklearn.ensemble: sklearn.ensemble is a module in the scikit-learn library that provides tools for implementing ensemble methods, which are machine learning techniques that combine multiple models to improve their performance.

- os: os is a built-in Python module that provides a way to interact with the operating system. It provides a simple and consistent interface to work with various aspects of the operating system, such as files, directories, processes, and environment variables.
- Here are some common use cases of os:
    1. File and directory operations: os provides functions for creating, deleting, renaming, and moving files and directories. For example, you can use os.mkdir() to create a new directory, os.listdir() to list the contents of a directory, and os.rename() to rename a file or directory.
    2. Path operations: os.path submodule provides functions for working with file and directory paths in a platform-independent way.

    3. For example, you can use os.path.join() to concatenate two or more path components, os.path.basename() to get the base name of a file or directory, and os.path.abspath() to get the absolute path of a file or directory.

    4. Environment variables: os.environ is a dictionary-like object that provides access to the environment variables of the current process. You can use os.environ.get() to get the value of a specific environment variable, and os.environ.setdefault() to set a default value if the variable is not defined.

    5. Process operations: os provides functions for working with processes, such as os.system() to execute a command in a subshell, os.fork() to create a new process, and os.kill() to send a signal to a process.

- Numpy: NumPy (Numerical Python) is a Python library that is used for scientific computing. It provides a high-performance multidimensional array object, as well as tools for working with these arrays. Here are some common use cases of NumPy.

## 4.2. MODEL ARCHITECTURE

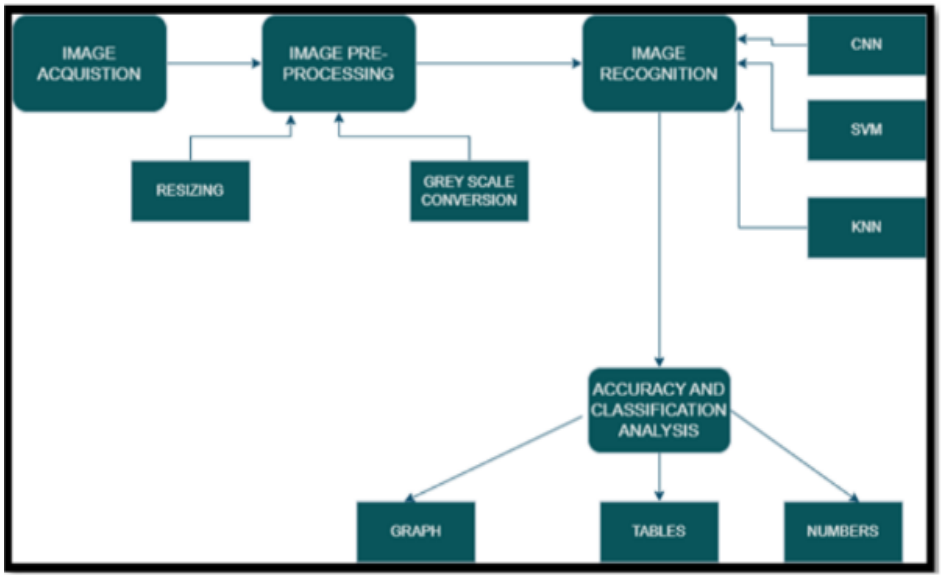
- The choice of model architecture for sculpture prediction would depend on the specific problem being solved and the characteristics of the input data.
- Assuming that the input data consists of images of sculptures, a common architecture used for image recognition tasks is a Convolutional Neural Network (CNN).
- A CNN typically consists of several convolutional layers that apply filters to the input image to extract features such as edges, shapes, and textures. These features are then passed through pooling layers to reduce the spatial dimensions of the output. The output of the convolutional and pooling layers are then fed into one or more fully connected layers, which perform classification on the features.
- To prevent overfitting, common techniques such as dropout and batch normalization can be used. Additionally, transfer learning can be used to leverage pre-trained CNNs that have been trained on large datasets such as ImageNet.

- In summary, a possible architecture for sculpture prediction using CNNs could consist of the following layers:

1. Input layer: Takes in the input image.
2. Convolutional layers: Apply filters to the input image to extract features.
3. Pooling layers: Reduce the spatial dimensions of the output.
4. Fully connected layers: Perform classification on the features extracted by the convolutional and pooling layers.

5 . Output layer: Outputs the prediction for the input image.

The number and size of the layers can be adjusted based on the complexity of the problem and the amount of available data.



**4.3. Software Description:**

REQUIREMENTS(S/W&H/W)

HARDWARE REQUIREMENT:

System : Ryzen5 ,5000 series

RAM : 8gb or above

Hard disk : 10GB OR above

Input : keyboard and mouse

Output : monitor or pc

SOFTWARE REQUIREMENTS:

OS : windows 11

Platform : jupyter notebook,goolgle colab

Program language : python

## 5.Results and Discussions

**Logistic Regression:**

```
from sklearn.metrics import confusion_matrix

cm=confusion_matrix(ytest,y_pred)

print("Confusion matrix:\n",cm)
```

Confusion matrix:

[[57 7]

[ 0 73]]

**KNN Regression:**

```
from sklearn.metrics import confusion_matrix
knns=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",knns)
```

Confusion matrix:

[[35 29]

[ 0 73]]

**NAIVE BAYES CLASSIFIER:**

```
from sklearn.metrics import confusion_matrix
nbb=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",nbb)
```

Confusion matrix:

[[56 8]

[ 1 72]]

**SUPPORT VECTOR MACHINE:**

```
from sklearn.metrics import confusion_matrix
SVMS=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",SVMS)
```

Confusion matrix:

[[62 2]

[ 0 73]]

**DECISION TREE CLASSIFIER:**

```
from sklearn.metrics import confusion_matrix
dtcs=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",dtcs)
```

Confusion matrix:

[[57  7]

[ 9 64]]

**RANDOM FOREST:**

```
from sklearn.metrics import confusion_matrix
rfcs=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",rfcs)
```

Confusion matrix:

[[57  7]

[ 0 73]]

**Accuracy scores of Machine learning algorithms:**

```
from sklearn.metrics import accuracy_score
accuracy_model = accuracy_score(y,model.predict(sc_x.transform(x)))
print("Logistic regression:",accuracy_model)
accuracy_nb = accuracy_score(y,nb.predict(sc_x.transform(x)))
print("navie bayes;",accuracy_nb)
accuracy_knn = accuracy_score(y,knn.predict(sc_x.transform(x)))
print("KNN:",accuracy_knn)
accuracy_SVM = accuracy_score(y,SVM.predict(sc_x.transform(x)))
print("Support vector machine:",accuracy_SVM)
accuracy_dtc = accuracy_score(y,dtc.predict(sc_x.transform(x)))
print("Descision tree:",accuracy_dtc)
accuracy_rfc = accuracy_score(y,rfc.predict(sc_x.transform(x)))
print("Random forest:",accuracy_rfc)
```

Logistic regression: 0.9872029250457038

navie bayes; 0.943327239488117

KNN: 0.7568555758683729

Support vector machine: 0.9963436928702011

Descision tree: 0.9707495429616088

Random forest: 0.9872029250457038

**Conclusion and Future scope:**

We got best accuracy with SVM and Random forest machine learning algorithms.

In conclusion, sculpture detection is an important and challenging task that has the potential to benefit the fields of art conservation, research, and education. The use of deep learning-based approaches has shown great promise in automating the process of sculpture detection, but there are still challenges that need to be addressed, such as the need for large and diverse datasets, the potential for bias in the training data, and the difficulty of detecting sculptures in complex or cluttered scenes.

Furthermore, as the field of computer vision continues to advance, there may be new techniques and approaches that can be applied to sculpture detection. For example, the use of generative adversarial networks (GANs) and other generative models may enable the creation of realistic 3D models of sculptures, even when only 2D images are available. Additionally, the use of attention mechanisms and other advanced deep learning techniques may improve the ability of sculpture detection systems to identify and analyze sculptures in complex and cluttered scenes.

Overall, sculpture detection is a rapidly evolving field with great potential for advancing our understanding and appreciation of sculptures around the world. With continued research and development, sculpture detection systems can become powerful tools for art conservation, research, and education.

**References:**

Thai Buddhist Sculpture Recognition System (TBuSRS) Chomtip Porn-panomchai, Member, IACSIT, Vachiravit Arpapong, Pornpetch Iamvi-setchai, and Nattida Pramanus IACSIT International Journal of Engineering and Technology, Vol.3, No.4, August 2011.

A. Dalara, C. Sindhu and R. Vasanth, "Entity Recognition in Indian Sculp-ture using CLAHE and machine learning," 2022 First International Confer-ence on Electrical, Electronics, Information and Communication Technolo-gies (ICEEICT), 2022, pp. 1-12, doi: 10.1109/ICEEICT53079.2022.9768565.

A. Saini, T. Gupta, R. Kumar, A. K. Gupta, M. Panwar and A. Mittal, "Im-age based Indian monument recognition using convoluted neural networks," 2017 International Conference on Big Data, IoT and Data Science (BID), 2017, pp. 138-142, doi: 10.1109/BID.2017.8336587.

G. Shalunts, M. Cerman and D. Albertini, "Detection of sculpted faces on buildingfacades," 2017 Asia-Pacific Signal and Information Processing As-sociation Annual Summit and Conference (APSIPA ASC), 2017, pp. 677-685, doi: 10.1109/APSIPA.2017.8282119.

Desai, P., Pujari, J., Ayachit, N. H., & Prasad, V. K. (2013). “Classification of archaeological monuments for different art forms with an application to CBIR”. 2013 International Conference on Advances in Computing, Commu-nications and Informatics (ICACCI). doi:10.1109/icacci.2013.6637332.

Kampel, M., Huber-Mörk, R., & Zaharieva, M. (2009). “Image-Based Re-trieval and Identification of Ancient Coins”. IEEE Intelligent Systems, 24(2), 26–34. doi:10.1109/mis.2009.29.