# What do you analyze in different phases of code? Explain both types of classification.

## Supervised Classification:

1. Load Libraries

Firstly, we have imported all the required libraries. We have used the Keras library for modeling and training. Also, command drive mount is used to mount google drive from where the data is loaded.

Graphical user interface, text, application

Description automatically generated

1. Load Data based on the labels

Now, a function is defined to load the images from the google drive path. Here 2 labels rugby and soccer are defined. Cv2 functions are used to convert BGR images to RGB and reshape the images based on the hardcoded image size value.

Text

Description automatically generated with low confidence

Then, the above function is called by passing the path as a parameter.

Graphical user interface, application

Description automatically generated

1. Process Data

Next, we perform Data Preprocessing and Data Augmentation. From the training data features and labels are gathered. Then the data is normalized and reshaped.

A picture containing shape

Description automatically generated

1. Data Augmentation

It is a way of generating more data points from the existing data. We have rotated images, zoomed in, horizontally flipped, and randomly shifted images horizontally and vertically.

Text

Description automatically generated

1. Building the Model

Here, the CNN model is defined with 3 convolutional layers. Also, a dropout layer is added after 3rd convolutional layer to avoid overfitting. Finally, we flatten the model, and the Dense function is used to classify the image based on the output from convolutional layers based on the activation function “relu” and “softmax”. we use “relu” in the hidden layer and the Softmax function is used in the last output layer.

Graphical user interface, text, application

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Now we compile the model using Adam as the optimizer and SparseCategoricalCrossentropy as the loss function. Here, the learning rate is 0.000001 which is very low for a smoother curve. Now the model is trained for 50 epochs. An epoch is made up of one or more batches, where part of the dataset is used to train the neural network.

Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated

1. Result

Finally, the results of the training and validation data accuracies and losses are plotted.

A picture containing graphical user interface

Description automatically generated

Chart

Description automatically generated

Finally, the classification report is displayed with precision and accuracy.

Graphical user interface

Description automatically generated with medium confidence

## Unsupervised Classification:

1. Load Data

Firstly, we have imported all the required libraries. We have used the Keras library to import the dataset minst. Load\_data() is used to load training and testing data. We check the ‘type’ and ‘shape’.

A picture containing background pattern

Description automatically generatedNow we plot a B/W image of the training data.

Graphical user interface, application

Description automatically generated with medium confidence

Now we print the y coordinates of training data. Now we check the minimum and maximum values of training data.

Background pattern

Description automatically generated

1. Processing the Data

Now training and test data are normalized, and the image input is divided by 255 so that input values are in the range of [0,1].

A picture containing background pattern

Description automatically generated

1. Building the model

We use MiniBatchKmeans to initialize the k-means model and fit the model to the training set.

Background pattern

Description automatically generated

The images are classified into clusters based on the similarity of pixel values. Each image is assigned a cluster label value given by kmeans.labels\_. Then we call the ‘retrieve\_info’ function and process it to get ‘number\_labels’ which denotes the number displayed in the image.

Graphical user interface, text, application, Teams

Description automatically generated

1. Result

Now we print the predicted and actual values for the first 20 values. Then calculate the accuracy score. Here, it is around 53%.

Background pattern

Description automatically generated

The below code is for optimizing the code.

The performance of the model is measured by calculating Inertia, Homogeneity score, and Accuracy score. We run the model for the random number of clusters.

**Note:**

**Inertia** is a measure of how internally coherent clusters are. The higher the number of clusters, the lower the inertia score.

**Homogeneity** is a measure of data points of a particular cluster belonging to a single class.

**The accuracy score** is the percentage of correctly predicted values.

1. Optimizing the Code

Graphical user interface, text, application, email

Description automatically generated

A picture containing shape

Description automatically generated

We notice that the accuracy score has increased as the number of clusters increases, Hence we use the test data with the highest number of clusters in the above run. Here the number of clusters is 256 and the accuracy score is 89.95%.

Graphical user interface, text, application

Description automatically generated

# Which differences did you identify in unsupervised and supervised image classification?

Below are a few main differences that have been identified between unsupervised and supervised image classification.

**Supervised image classification:**

1. In supervised classification, training samples are selected and classified based on chosen labels. The training samples are key as they determine the class of each pixel in the image. Here 2 groups of images soccer and rugby were given in the code for training and testing purposes.
2. Supervised training requires a teacher to classify training data among the labels.
3. It maps data to the information class. Here the images are mapped to the rugby and soccer classes.
4. In supervised image classification, the CNN model is used for training the data.
5. Accuracy score does not depend on the number of samples. As the sample size increases “statistically significant” of the result will increase.

**Unsupervised image classification:**

1. Unsupervised classification generates clusters based on similar characteristics in the image and classifies each cluster. There are no sample training sets in unsupervised classification. Here, the data has been divided into clusters using the k-mean classification method.
2. Unsupervised training happens without a teacher, data is classified among unlabeled bins/clusters.
3. Data space is explored to discover statistical similarities in the images to classify them among the clusters.
4. In Unsupervised image classification, the K-means model is used for training the data.
5. As the number of clusters increases, the accuracy score also increases.