BIGDATA AND MACHINE LEARNING

Sakshi Patel

Assignment 1: Raw Data to Feature Space

I. Task 1

Demonstrate the knowledge of basics
Summary of Chapter 1

A. Data Science and Information Systems

Data Science is the management and analysis of data sets, the extraction of useful information, and the understating of the systems that produce the data. Data science is applied to many real-world systems, such as network security (intrusion detection) and climate change monitoring. These systems generate vast amounts of complex data. Handling this kind of ”big data” can be challenging because it is often large, complex, and unstructured.

B. Big Data Paradigm

There is two important terms, data and knowledge. Data is the hidden digital facts. Data could be labeled or not labeled. In the labeled data digital facts are not hidden and used for training the machine-learning techniques. In the unlabeled data, the digital facts are hidden and used for testing or validation. Knowledge is learned information acquired from the data. Monitoring system needs three operations, called physical, mathematical and logical operations. Physical operations involved steps of data capture, data storage, data manipulation, and data visualization. In mathematical operations, mathematical and statistical techniques and tools required for the transformation of data into knowledge. Logical operations describe the logical arguments, justifications and interpretations of the knowledge.

To understand the distinction between the big data and regular data, we need to understand three parameters, n, p, and t of a system. The parameter n represents the number of observations captured by system at time t, the parameter p represents the number of features.

C. Machine Learning Paradigm

The machine learning paradigm provides solutions for analyzing big data. The chapter explains how machine learning, through both supervised and unsupervised learning methods, can derive insights from large datasets.

Machine learning is about the development of mathematical models and algorithms to learn from data. The classification is also called supervised learning, which requires a training (labeled) data set, a validation data set and a test data set. The training data set helps find the optimal parameters of a model, validation data set helps avoid overfitting of the model, and the test data set helps determine the accuracy of the model.

In machine learning, the term modeling means mathematical and statistical modeling of data and algorithm is to derive a model.

In supervised learning, the classes are known and class boundaries are well defined in the given data set. So the classification is also called supervised learning. In unsupervised learning, classes or class boundaries are not known, hence the class labels themselves are also learned, and classes are defined based on this. Hence, the class boundaries are statistical and not sharply defined, and it is called clustering.

A key challenge in big data science is determining whether data is truly ”big” or just complex enough to be handled by traditional tools. Another challenge is finding the right techniques to manage the ever-growing scale of big data efficiently.

This chapter introduce some modern field of data science and current progress in data science. It’s focus on problem space and solution space for the big data analytics. Also introduce with important elements of data science: Data, Knowledge and Responses.

Summary of Chapter 2

A. Big Data Analytics

Big Data Controllers: There are three important controllers: Class characteristics, feature characteristics and observation characteristics. Class characteristics determines variety, feature characteristics determines size and dimensions and observation characteristics determines size and volume. Number of features p and number of observation n, together define the characteristics of dimensionality. If n¡p, then data set is said to be high dimensional.

Big Data Problems: Controller class contribute to unpredictability problems of big data, controller features contribute to complexity problems and controller observations contribute to managing, processing and analyzing the data.

Challenges: It's highlights the significant challenges in big data analytics. The problems caused by the controller class can impact the performance degradation of the classification techniques. The problem caused by the controller feature challenge the reduction of dimensionality and the storage and computing power. Controller observation cause the challenge in the data processing, storage requirements, and communication issues when the data are distributed.

Solutions: Solutions technology such as Linux, Hadoop, MapReduce and Programming helps to addressing the challenges associated with the big data controllers. The challenges associated with the big data controllers involve techniques to solve the problems with respect to their speed, complexity, unpredictability, (un)manageability and scalability.

B. Big Data Classification

Representation Learning: The representation learning techniques useful for understanding and shaping the data. It's mainly focus on the big data controller feature, and its goal is the feature selection. So it's contributes to the dimensionality reduction objectives in machine learning. The main goal of representation learning is the feature selection.

Distributed File Systems: It's suitable for big data analysis, management and processing. Like Hadoop's HDFS, allow efficient storage and parallel processing of large datasets by distributing the data across multiple machines.

Classification Modeling: Class characteristics defined by imbalanced, incomplete and inaccurate data. Imbalanced data means the classes are not balanced. Incomplete data means the incomplete features (missing values in some of features). Inaccurate data means the observation are not correct. It means some of the observations are not correctly labeled. The error characteristic defined by the approximation, the estimations and the optimization errors. Estimation error may be defined as the differences in the models derived from the data sets of different sizes. Approximation error may be defined as the differences in the models derived from the parametrized models assumed. Optimization error may be defined as the differences in the algorithms used to derive the models. Domain characteristics defined by dimensionality, sparsity and subspace. It's show the relationship between the three characteristics. The new space with fewer features is called the subspace.

Classification Algorithms: Classification algorithms mainly involve machine learning processes: training, validation, and testing. Training phase provide algorithm to train the model. Validation phase provide algorithm to validate the effectiveness of the model and testing phase provides an algorithm to test if the trained and cross validation works using another data set.

C. Big Data Scalability

Scalability problem created by uncolorable and continuous growth in the features. It occurs in high-dimensional system. A large number of feature does not mean the data is high dimensional.

High-Dimensional Systems: As the number of features (dimensions) increases in big data, it becomes challenging to manage and analyse the data efficiently. A data set is high dimensional only if the number of features (p) of the data set are larger than the number of observations (n) in the data set.

Low-Dimensional Structures: This topic explain how data can be transformed into low-dimensional structures to simplify analysis and improve scalability. Hashing techniques used to create low-dimensional subspecies.

Summary of Chapter 3

Big Data Analytics involves extensive analysis of large datasets to uncover hidden patterns, trends, and insights. The objective of this chapter is to illustrate some of the meaningful changes that may occur in a set of data when it is transformed into big data through evolution. Split-merge-split framework is developed to make this objective practical and interesting.

A. Analytics Fundamentals

Analytics in big data is to reveal data properties through statistical measures like counting, mean, standard deviation, variance, covariance, and correlation. These numerical measures used to understand the underlying structures in the data, aiding classification and pattern detection.

In addition to statistical measures, graphical tools such as scatter plots, histograms, and graphs are important for visualizing data properties. These tools used to identify patterns that may not be evident through numerical analysis alone. Both numerical and visual techniques are used to fully understand the characteristics of large datasets and optimize machine-learning algorithms. The choices of data sets and the way they are handled to explain the big data classification are very important.

B. Pattern Detectors in Big Data

In big data analytics we identify the patterns through statistical and graphical measures. This chapter introduced several core statistical measures, such as:

1) Counting: Counting play major role in addressing imbalanced data problems in big data classification.

2) Mean: Provides the average value of a feature or observation.

3) Variance: Shows the spread or dispersion of data points.

4) Covariance: Measures the relationship between two variables.

5) Correlation: Quantifies the strength and direction of a relationship between variables.

Along with statistical measures, graphical measures, such as histograms, skewness and scatter plots, help in understanding the visual patterns within data. These tools are especially useful when dealing with high-dimensional data, where direct interpretation of raw numbers can be complex.

C. Patterns of Big Data

Pattern evolution is main goal in this chapter. Pattern evolution is a natural phenomenon in big data environment. As data transforms over time, patterns within the data may propagate, deform, or evolve. The evolution of patterns increases the complexity of the data, and, therefore, the development of supervised learning models and algorithms for big data classification is difficult. Detecting these changes requires the application of statistical techniques, such as standardization and normalization, to bring out hidden patterns. Normalization is the process of scaling individual features to a similar range, typically between 0 and 1. Standardization is the process of transforming features to have a mean of 0 and a standard deviation of 1. normalization, and standardization are essential concepts in machine learning that help ensure the effectiveness, stability, and interpretability of models. These processes adjust data distributions, making it easier to apply machine learning techniques effectively.

One more concept introduced is the idea of data expansion modeling—a method used to manage the growing complexity of data by expanding feature spaces. Data expansion can provide evolution and the deformation of patterns to study the data characteristics.

Additionally, dealing with deformation of patterns: imbalanced, incomplete, or inaccurate data. Also understand classification errors characteristics called approximation error, estimation error, and optimization error. Approximation is defined as the error in the parametrized model used. Estimation is defined as the error in the parameters used. Optimization is defined as the error in the learning algorithms used. Optimization error can be obtained by comparing the estimation errors derived using different algorithms.

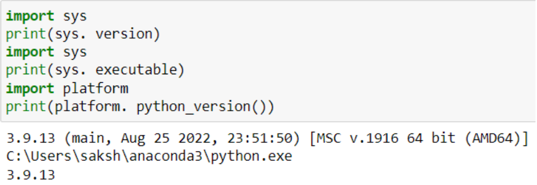
D. Low-Dimensional Structures

Low-Dimensional structures focus on the meaningful low-dimensional structures. It's display meaningful patterns and demonstrates the usefulness of data reduction to low dimensions and data interpretation.

II. Task 2

Build your programming environment!

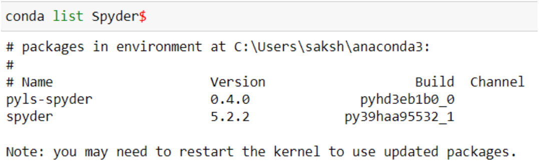
In this assignment, the Python programming language was utilized. Fig. 1 illustrates the installation of the Python environment. Fig. 2 and Fig. 3 display screenshots of the Spyder IDE and Jupyter Notebook, respectively. Fig. 4 presents the command used to install the OpenCV library, which was required for the implementation of computer vision tools in this assignment.



```
import sys
print(sys. version)
import sys
print(sys. executable)
import platform
print(platform. python_version())
```

```
3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
C:\Users\saksh\anaconda3\python.exe
3.9.13
```

Fig. 1. Python environment installation

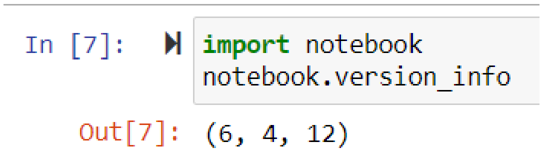


```
conda list Spyder$
```

```
# packages in environment at C:\Users\saksh\anaconda3:
#
# Name                    Version                   Build  Channel
pyls-spyder               0.4.0              pyhd3eb1b0_0
spyder                    5.2.2            py39haa95532_1

Note: you may need to restart the kernel to use updated packages.
```

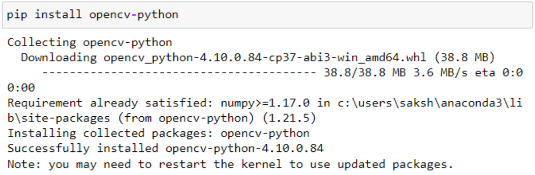
Fig. 2. Spyder IDE



```
In [7]:  import notebook
         notebook.version_info

Out[7]:  (6, 4, 12)
```

Fig. 3. Jupyter Notebook



```
pip install opencv-python
```

```
Collecting opencv-python
  Downloading opencv_python-4.10.0.84-cp37-abi3-win_amd64.whl (38.8 MB)
     ---------------------------------------- 38.8/38.8 MB 3.6 MB/s eta 0:0
0:00
Requirement already satisfied: numpy>=1.17.0 in c:\users\saksh\anaconda3\li
b\site-packages (from opencv-python) (1.21.5)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.10.0.84
Note: you may need to restart the kernel to use updated packages.
```

Fig. 4. Command used to install OpenCV

III. Task 3

Generate / download a dataset of bird images!

In this assignment I selected three birds images: Cardinal, Sparrow and Red-Bellied Woodpecker. Fig. 5 is a color image of Cardinal, Fig. 6 is a color image of Sparrow and Fig. 7 is color image of Red-Bellied Woodpecker.

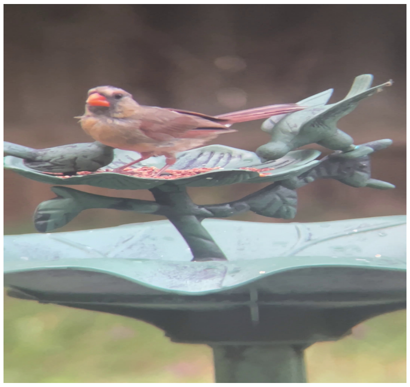


Fig. 5. Cardinal (image0)

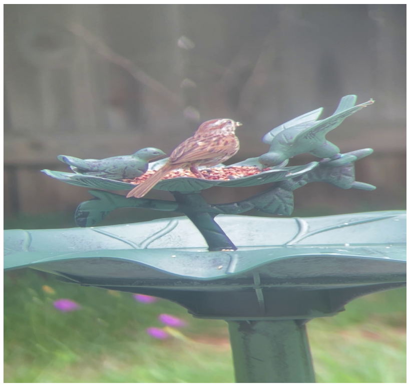


Fig. 6. Sparrow (image1)

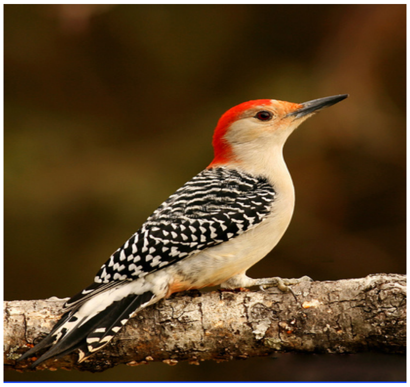


Fig. 7. Red-Bellied Woodpecker (image2)

IV. Task 4

Write a simple code to read your selected images and display them on the programming environment! Before we start with programming first we need to import some library such as pandas, matplotlib, numpy, cv2. In this task we need to read these three color images and display it in RGB channel images. After that we converted that image into grayscale and display the dimensions of all this three images.

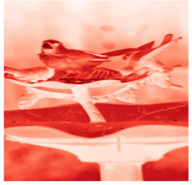


Fig. 8. Red_image0

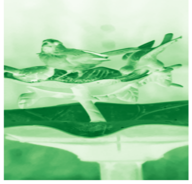


Fig. 9. Green_image0

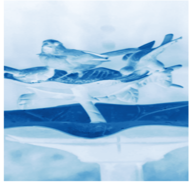


Fig. 10. Blue_image0

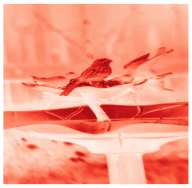


Fig. 11. Red_image1

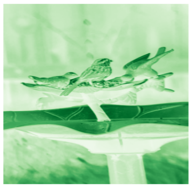


Fig. 12. Green_image1

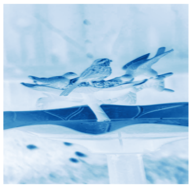


Fig. 13. Blue_image1

V. Task 5

Resize the images to reduce their dimensions! In this task, we need to resize the grayscale images such that the output dimensions are divisible by 16, while preserving their original aspect ratios as closely as possible.

The aspect ratio is the relationship between an image's width and height. It has been specified that the grayscale images must be resized to a height of 256 pixels, and the aspect ratio should be maintained when calculating the width of the target resized image. The new width of the resized image is calculated using the following formula:

$$\text{aspect\_ratio} = \frac{\text{original\_w}}{\text{original\_h}} \tag{1}$$

$$\text{new\_width} = \text{target\_height} \times \text{aspect\_ratio} \tag{2}$$

Where:
- original_w is the original width of the image,
- original_h is the original height of the image,
- target_height is the desired height for the resized image,
- aspect_ratio is the ratio of the width to the height of the image,
- new_width is the calculated width that maintains the original aspect ratio of the image.

After calculating the new width of the images, we need to ensure that the new height and width

Fig. 14. Red_image2

Fig. 15. Green_image2

of each image are divisible by 16. Therefore, the resulting resized image dimensions are as follows:
image0: (256, 144)
image1: (256, 144)
image2: (256, 208)

Fig. 16. Blue_image2

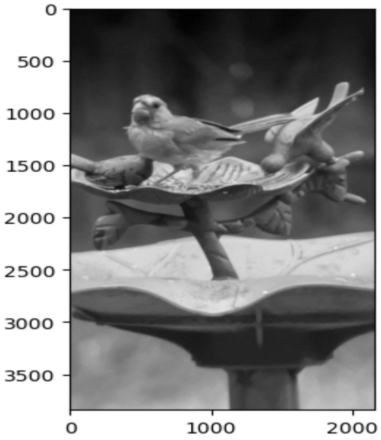


Fig. 17. Grayscale_image0

VI. Task 6

Generate block-feature vectors!

In this task, I first created blocks of 16x16 pixels from the image. In Fig. 20, we can see an example of one such 16x16 pixel block. Next, I flattened the block into a 1D vector of size 16x16 = 256. After creating the feature vector, which has a size of 256, I appended labels to each feature vector with values 0, 1, and 2. The label for image0 is 0, for image1 it is 1, and for image2 it is 2.

After generating the feature vectors for each image—where each row represents a feature vector. I combined all the feature vectors into one dataframe. Finally, I generated a spreadsheet named feature_vectors.csv, which stores the feature vectors for all the images (image0, image1, and image2).

VII. Task 7

Generate sliding block-feature vectors!

This task is about to create new feature vector with the idea of sliding block. The idea behind it is move a block of fixed size (in this case 16x16 pixels) across an image, extracting and flattening each block into 1D feature vector. The function slides the window across the image using a step size, which determines the overlap between consecutive blocks.
Block size: Defines the dimensions of the window (16x16 pixels in this case).
Step size: Controls how much the window moves with each step (8 pixels in this case).
Flattening: The block is flattened into a 1D array of size 256 (16x16).
The sliding window block plays a crucial role in breaking down images into smaller block with overlap between consecutive blocks, allowing for effective feature extraction that captures local information. It is useful in tasks such as image classification or object detection.
After creating sliding window feature vectors for each images with the labels I combined all the feature vectors into one dataframe. Finally, I generated a spreadsheet named sliding_window_feature_vectors.csv, which stores the sliding window feature vectors for all the images (image0, image1, and image2). Fig. 21 and Fig. 22 illustrate the concept of generating block feature vectors and sliding block feature vectors.

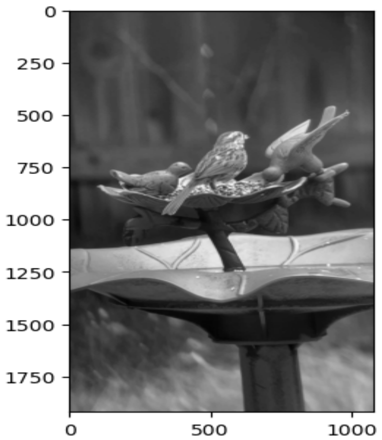


Fig. 18. Grayscale_image1

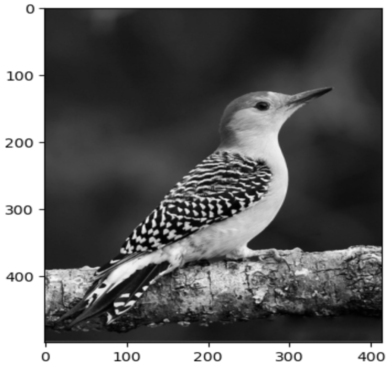


Fig. 19. Grayscale_image2

VIII. Task 8

Derive statistical descriptors!

In this task we need to extract some statistical information (e.g., number of observations, dimension of the data, mean of each feature, etc.) from datasets.Table 1 show the statistical values for image0, image1 and image2.

Also presented the visual representation of image0, image1 and image2 in histogram and scatter plot.

Is the dataset imbalanced, inaccurate, or incomplete?

Imbalanced Data:Imbalanced data means the classes are not balanced. That means one or more classes have significantly fewer or more instances than the others.

I did the code for calculating class distribution and I get the below class distribution for label 0,1 and 2.

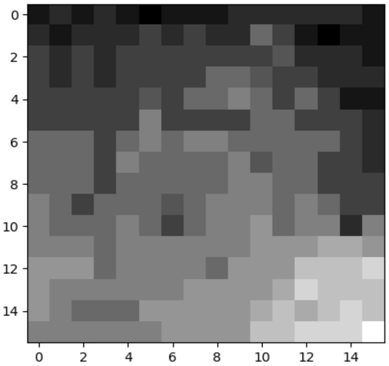


Fig. 20. Block of 16x16 pixels

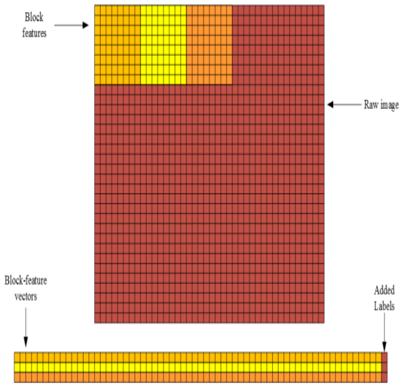


Fig. 21. Generating block feature vectors

Class 0 : 144
Class 1 : 144
Class 2 : 208

Class 0 and Class 1 have the same number of instances (144 each). Class 2 has 208 instances, which is 64 more than Class 0 and Class 1. It could be considered slightly imbalanced because there is some variation. The ratio of the largest to smallest class is about 1.44 : 1. This level of imbalance is relatively mild and may not cause significant problems for many machine learning algorithms

Inaccurate Data: Inaccurate data means the observation are not correct. It means some of the observations are not correctly labeled.

Is it a trivial data or possibly a big data?

If the data has complex structure(mixture Gaussian Model), classification of image is difficult and patterns are hidden in the plot then that data is called big data.

Above is the 2D ploting of feature 1 and 2 of image0,image1 and image2. Which show that it's

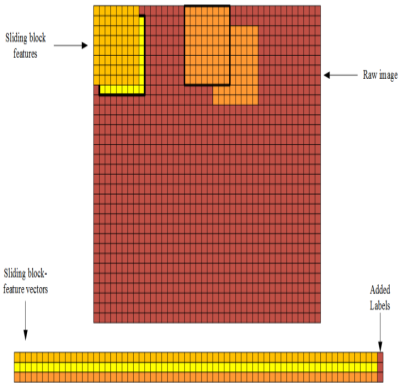


Fig. 22. Generating sliding block feature vectors

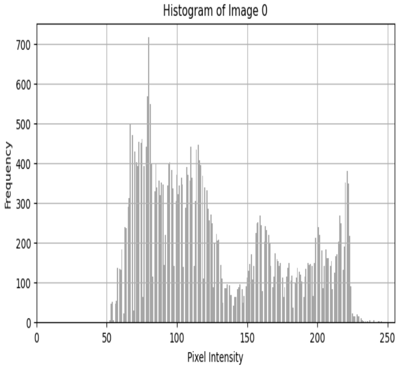


Fig. 23. Histogram of Image0

TABLE I
STATISTICAL OVERVIEW OF IMAGE 0, IMAGE 1, AND IMAGE 2

Statistic	Image 0	Image 1	Image 2
Number of observations	36,864	36,864	53,248
Dimension	(256, 144)	(256, 144)	(256, 208)
Min	52	74	0
Max	246	254	255
Mean	126.809	134.703	65.974
Std Dev	49.264	34.377	55.733

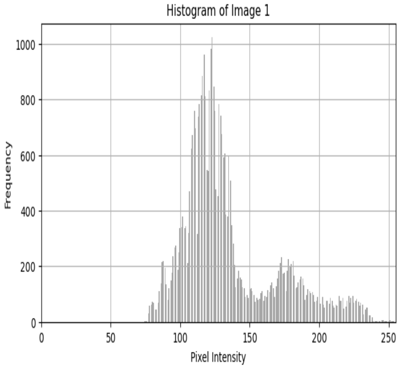


Fig. 24. Histogram of Image1

very difficult to find the patterns in this plot. So we can say that it's big data. However, we can find some hidden patterns using standardization, normalization, linear transformation, and orthogonalization over the feature variables.

Are they high dimensional?

If n ¡ p, then the data set is said to be high dimensional where n is number of observations and p is number of features. Accoding to this Image0, Image1 and Image2 all are not high dimensions.

Do you need to standardize?
Standardization is the process of transforming features to have a mean of 0 and a standard deviation of 1. It centers the data around zero and scales it to have unit variance. After calculating mean and standard deviation, the standardized means for image0, image1, and image2 are very close to zero, as approximately 0. This indicates that the average pixel value across each of the images has been shifted to the center of the distribution. **Do you need to standardize?**
Normalization scales the data between 0 and 1 or another range. It ensures that all features have a similar influence on the model, regardless of their original scales. Normalization would involve scaling image0, image1 and image2 to a common range, such as [0, 1]. This ensures that both features contribute equally to the model, regardless of their original magnitudes. Normalization prevents features with larger scales from dominating the model's training process and biases. It ensures that all features are treated equally and prevents numerical instabilities during optimization. Normalized features lead to more stable and efficient learning algorithms.

IX. TASK 9

Construct a feature space!
In this task, I first created spreadsheets for each image, containing their individual feature vectors, named: image0.csv, image1.csv, and image2.csv. Following the task requirements, I then created a feature space by merging the feature vectors of

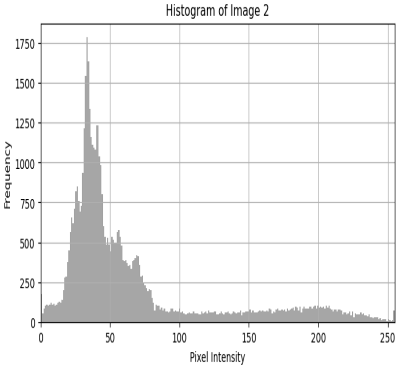


Fig. 25. Histogram of Image2

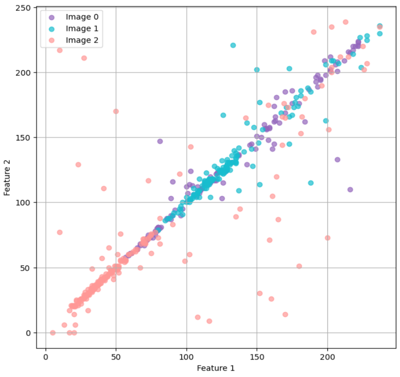


Fig. 26. Scatter plot of image0, image1 and image2

image0 and image1, and named it image01.csv, with a dimension of (288, 257), where the 257th column contains the label. Similarly, I merged the feature vectors of image0, image1, and image2 to create a feature space for all three images, named image012.csv, with a dimension of (496, 257).

For the third sub-task, I randomized the placement of the feature vectors in both image01.csv and image012.csv and renamed them as random_merged_image01.csv and random_merged_image012.csv, respectively. The purpose of this randomization is to prepare the data for training a machine learning model.

X. Task 10

Display subspace!
In this task we need to select the feature and plot the two-dimensional and three-dimensional. Fig. 29 show the two dimensional feature space for first and second feature and Fig. 30 show the three

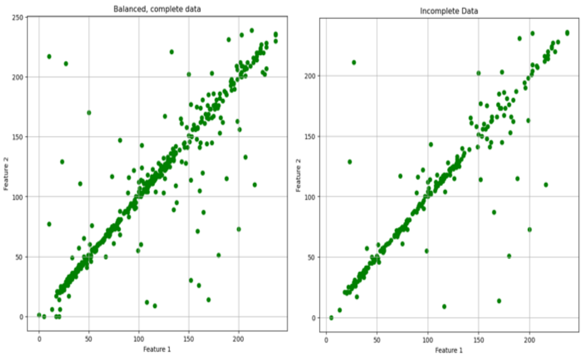


Fig. 27. Incomplete data

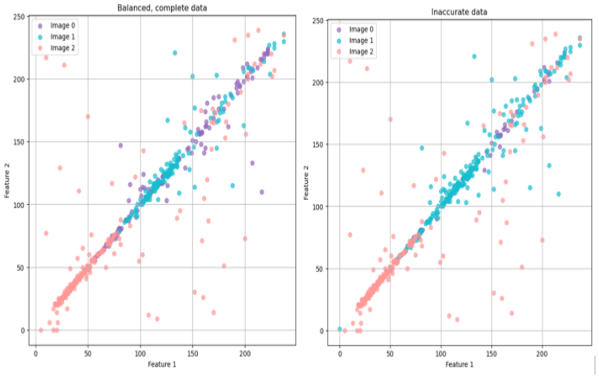


Fig. 28. Inaccurate data

dimensional feature space for first, second and third feature.

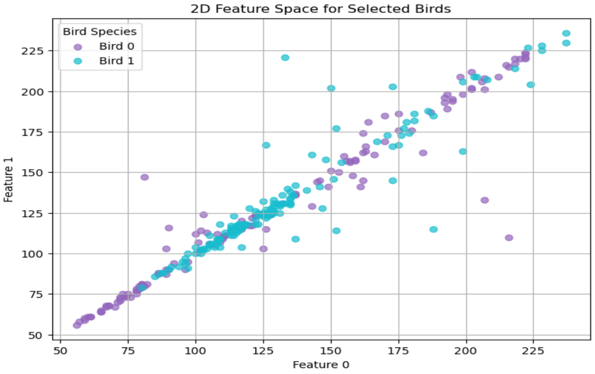


Fig. 29. 2D Feature Space

2D Feature Space: The 2D scatter plot shows a significant overlap between the two bird species (Bird 0 and Bird 1) based on the chosen features (Feature 0 and Feature 1). This overlap indicates that the selected features alone may not be enough to

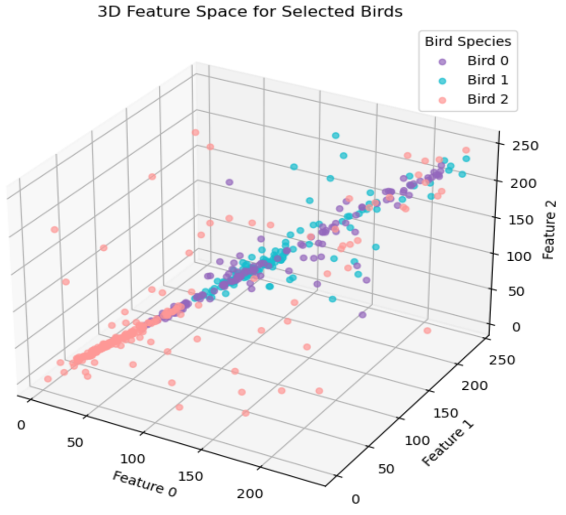


Fig. 30. 3D Feature Space

distinguish between the two bird species effectively. The scatter points seem to align along a diagonal axis, suggesting that there is some correlation between Feature 0 and Feature 1. Both bird species follow a similar linear pattern, further complicating classification based on these two features alone. The plot shows that in some regions, especially in the central portion of the scatter plot, it would be difficult to classify the bird species based on these features due to high overlap.

3D Feature Space: Introducing a third feature (Feature 2) slightly improves the separation of the bird species. While Bird 0 and Bird 1 still overlap, there is better visual differentiation compared to the 2D plot. The inclusion of Bird 2 (the green points) adds a new class, which is more spread out and distinct. In 3D space, you can observe that Bird 2 is more separable due to the variation in Feature 2, and its points are more dispersed compared to the other two bird species. The added dimension helps in visualizing a bit more variance among the species, especially for Bird 2. Despite the third dimension, Bird 0 and Bird 1 still exhibit significant overlap, which indicates that additional or more discriminative features might be needed for clearer separation between these two classes.

XI. Task 11

Make appropriate changes to your Python code such that it can read any number of images from a folder that consists of many similar images, generate a feature space/s, and a spreadsheet/s for the feature spaces!

In this task, I modified my code to read all images from the folder and create the feature space for each of them and generated spreadsheet with name folder_feature_vectors.csv.

XII. Task 12

A. Impact of Block Size on Vector Count and Feature Space Dimensionality

The choice of block size plays a key role in shaping the dimensionality of the feature space and the number of vectors that represent the data. When transforming raw data into a feature space, block size directly affects how the data is divided, influencing both the complexity and the performance of classifiers. This section explores how varying block sizes impact the feature space and the classifiers' functionality within it.

B. Dimensionality of the Feature Space

The dimensionality of the generated feature space is highly dependent on the selected block size. Larger block sizes tend to capture more complex, detailed representations of the data, revealing finer patterns. However, this also results in higher dimensionality, which can cause problems such as increased computational demands and the ”curse of dimensionality” (where too many dimensions complicate analysis).

Conversely, smaller block sizes produce simpler, lower-dimensional representations, which may be easier to compute but can miss important details in the data.

Choosing the right block size involves finding a balance between capturing enough detail for the task (such as classification or regression) and ensuring that the computational load remains manageable. The choice should be informed by the characteristics of the dataset and the specific goals of the analysis.

C. Number of Vectors in the Feature Space

The number of vectors that represent the data in the feature space is also influenced by block size. Larger blocks cover more area, resulting in fewer but broader vectors. This reduces the total number of vectors but may fail to capture finer variations in the data. Smaller blocks, on the other hand, increase the number of vectors, allowing for a more detailed representation of the data but potentially introducing noise or redundancy.

The number of vectors is important because it affects the classifier's ability to interpret and process the data. Too few vectors may cause the classifier to miss subtle patterns, while too many vectors may lead to overfitting or inefficiencies due to noise.

D. Impact on the Classifier

The choice of block size has a significant impact on the classifier's performance. A classifier trained on a high-dimensional feature space (resulting from larger block sizes) may be more sensitive to changes in the data and can capture finer details, but it is also at higher risk of overfitting. In contrast, a classifier operating in a lower-dimensional feature space (from smaller block sizes) may be more stable but might overlook important nuances in the data.

Additionally, the block size affects more than just model complexity; it also influences computational efficiency and generalization ability. Selecting the optimal block size ensures that the classifier can balance accuracy and efficiency, aligning with the task's goals and the data's inherent characteristics.