SEINet CACTACEAE Distribution and Similarity Detection

Kasturi C Adep
Arizona State University
Tempe, Arizona, USA
kadep@asu.edu

Shafquat Bakhtiyar
Arizona State University
Tempe, Arizona, USA
sbakhtiy@asu.edu

Dhriti Anil Shah
Arizona State University
Tempe, Arizona, USA
dshah25@asu.edu

Shrimangal Rewagad
Arizona State University
Tempe, Arizona, USA
srewaga1@asu.edu

Omkar Sandeep Vedak
Arizona State University
Tempe, Arizona, USA
ovedak@asu.edu

ABSTRACT

Broad range of cactus species found in the United States and Mexico have resulted in the creation of a complex and multi-organizational dataset. Visualizing this data which is a subset of the SEINet CACTACEAE dataset presents a challenge due to its size. And so, we attempt to build an application that best presents the trends in the underlying data. We particularly use different interactive graphs such as heat maps, network and bubble charts, etc. Moreover, this dataset has been used to develop an image label prediction module. Based on features extracted by processing images in the data, we use unsupervised clustering to find similar images.

KEYWORDS

Data visualization, SEINet CACTACEAE, Image similarity, Unsupervised nearest neighbors, Local Binary Patterns, Chi-squared distance, Histogram of Gradients, Euclidean distance.

1 INTRODUCTION

Adding visuals to long boring text has been a very common technique to generate interests in the viewers. One possible reason people find cacti to be boring plants is the lack of visual appeal. Humans are visual creatures by birth. Images, videos, interactive charts help learning and understanding in our minds. And so they will definitely feel uninterested to browse through a website full of long tabular data because humans are visual creatures by birth. We use these strengths of data visualization to address a similar problem, where we tell our story of the cactus data in a fun, interactive and interesting way.

Image prediction is a highly used technique. Even we use the technology in phones to detect and identify objects. The challenge in building such systems however is the availability of labelled data. The SEINet dataset is a large labelled set with thousands of cactus images of different species. Building an image predictor is thus one of best uses of this dataset. There are many enthusiasts and scientists who can use such tools to identify and label unknown plant images. And so, we propose an application that provides a list of similar plants to your input.

2 MOTIVATION

Cacti are probably deemed as one of the most monotonous categories of plants. Anyone with no botanical background could easily assume all cacti to look alike. Our aim is to break this stereotype and present users with an application that calls attention to the widespread and magnificence of these tough species of plants. Each one of the visualizations highlight a characteristic of the cactus plant. The idea is to present a one-stop shop for the entire dataset. Anyone viewing this application will get information related to biological and geographic attributes of the species.

Finally, we want users to be able to identify the species of a cactus from an image captured by them. For someone living in Arizona, this application could be used to identify a random cactus growing in a person’s backyard. While there are applications that predict species for all plants, our application entirely focuses on cacti. We estimate that our application should provide better results for cactus than others as features are solely extracted and compared for one broad species of plants. And so, we build an image predictor that presents top six predictions based on the similarity of the image with images in our dataset.

3 VISUALIZATION DESIGN

3.1 System implementation

This section talks about the architecture of the application. The website is divided into two main parts. The first is the dashboard, that has six different visualizations. The number of graphs was decided on the basis of the data attributed we wished to cover. Based on our analysis we identified that to provide a complete overview of the dataset the visualizations must cover both biological and geographical attributes of cacti. A couple of graphs were built by aggregating data for different attributes and showing trends across them. The following sections discuss the choices and reasons behind chart selection.

The second part of the application is the intelligence of our system. This involves uploading a file that the user would like to identify using the SEINet dashboard project. Following the upload, the file is sent to the backend api, that extracts features for the input. These features are then used for comparison with existing images in the datasets to identify most suitable labels. Section 3.1.3 discusses the algorithms used for implementing the image predictor module.

3.1.1 Data collection

All of our data was acquired from the official SEINet website. To collect the necessary data, we build a web scraper that parses the html from the website to extract the relevant details. The SEINet website allows you to filter data based on species (CACTACEAE) and presents a table of plant details related to the name, location, habitat, taxonomy and image URIs.

The web scraping module uses Beautiful Soup, a Python library for parsing HTML files, traverses through anchor tags for each row in the table to extract the plant id and image resource locator. All data is exported to a csv file.

All the images necessary for providing details and the image detection module need to be downloaded using a second script. This script reads image resource locators from the previously generated csv, downloads the file from the web and assigns a unique identifier, that corresponds to the details in the data file.

3.1.2 Data cleaning and pre-processing

The data extracted from the SEINet portal had 75000+ records. Image feature extraction and distance calculation requires GPUs and faster memory. Due to limited hardware capacity we decided to limit the data to North America, particularly the countries of USA and Mexico.

Next we eliminated records with missing values location. While it was possible to obtain the missing information from other sources we decided to maintain the integrity of the data by collecting data from a single source. Another benefit of eliminating records with missing values was that this served as a filter for reducing the number of records. As mentioned earlier our computing resources were limited and reducing the size of data was crucial to build a functional prototype.

Each species is associated with a taxonomy id and on observing the data it was found that there were multiple records for each id. On further analysis it was established that while the species was the same, the image associated with the record was unique. We used this to improve the image predictor module and maintained 3 different records to improve the accuracy. This would increase the chance for our algorithm to identify the correct label even if the uploaded image was different in terms of size or angle.

3.1.3 Similarity matching algorithm

The main aim of image matching is to find similarity between images using feature vectors. Here, texture analysis is used to find the feature vectors. Linear Binary Patterns (LBP) is a non-parametric method which is tolerant to illumination and gray-scale changes [1]. While, Histogram of Gradients (HoG) characterizes the apparent objects and shape by evaluating the distribution of the gradient of the density or the edges directions [2]. For image matching Local Binary Patterns(LBP) and Histogram of Gradients(HoG) texture descriptors were adopted. In order to find the distance, metrics such as Euclidean Distance and Chi Squared Distance are used. The algorithm covers following steps -

Step 1: Load the color image and convert to Grayscale

Step 2: Extract Feature Descriptor using LBP and HoG

Step 3: Calculate Euclidean and Chi-Squared distance between the feature descriptors of uploaded image and images stored in the database.

We calculated feature descriptors using two famous texture analysis algorithms.

LBP: LBP is a very efficient and powerful texture operator; it labels the pixels of an image by thresholding the neighborhood of the pixel in a binary format. Due to its discriminative power and computational simplicity, LBP texture operator has become a popular approach in various applications in image processing [6]. The technique works as explained below.

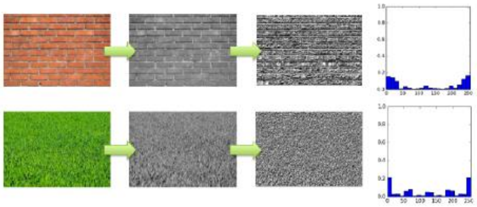


Figure 1: LBP Mask: Bikramjot Singh Hanzra (May 30, 2015) Texture Matching.Retrieved from http://hanzratech.in/2015/05/30/local-binary-patterns.html

1. Divide the window into a fixed number of cells.
2. Calculate LBP Mask.
3. Computing histograms over each cell.

4. Collect all the histograms and combine into a single feature descriptor

Calculation of LBP Mask using Python Library:
A built-in library function is used for calculating texture descriptors. Number of points, radius and method are important parameters.
a. The number of points ‘n’ in a circularly symmetric neighborhood to consider (thus removing relying on a square neighborhood) [5].
b. The radius of the circle r, which allows us to account for different scales [4].
c. A local binary pattern is considered to be uniform if it has at most two 0-1 or 1-0 transitions. For example, the pattern 00001000 (2 transitions) and 10000000 (1 transition) are both considered to be uniform patterns since they contain at most two 0-1 and 1-0 transitions. The pattern 01010010) on the other hand is not considered a uniform pattern since it has six 0-1 or 1-0 transitions [5].

HoG: In the HoG algorithm we count occurrences of gradient orientation in localized portions of an image. These localized spaces are small uniformly spaced cells and use local contrast normalization to increase the accuracy. HoG considers edge detection as its priority and one direction of each pixel rather than all the 8 directions in a pixel.

Calculate Gradients using Python Library:
A Python package called skimage (scikit-image) is used to calculate the HoG descriptors. Parameters of HoG descriptors are orientations, pixels per cell, cell per block and block-norm. pixel per cell defines how much course or fine the feature descriptor image will be. More the value of pixels per cell represents a coarse image. Number of bins in a histogram are defined by orientations. To keep a track of contrast in gray-scale images, a number, cell per block is used to normalize gradient values. The cells are grouped together into larger groups of blocks. Normalization makes image illumination more resistant to changes. All the histograms obtained are appended and normalized. block-norm parameter specifies L1-norm or L2-norm is used.

Storing and Retrieving images using Sqlite:
Feature descriptors of each image are calculated and saved in a Sqlite database to improve the efficiency of storage and retrieval of image.

Image Matching using Distance Metrics(d):
Two distance measures are used to find the similarity distance. Lesser the distance more similar the image is with the query image.

Euclidean Distance:

Euclidean Distance Metrics:

$$d = \sqrt{\sum (a-b)^2}$$

Chi-Squared Distance:

Chi-Squared Distance Metrics:

$$d = 0.5 * \sum (observed - expected)^2 / (observed + expected)$$

Similarity: Similarity between the image features can be calculated using the formula -

Similarity = (1 / 1 + d) * 100

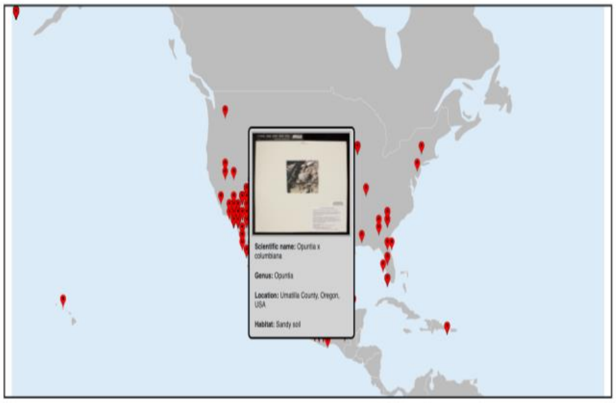
3.1.4 Technologies used

The frontend of the application is developed using D3.js and standard JavaScript in conjunction with HTML and CSS for styling. On the backend we use the Flask framework to host a Python API. Data collection, aggregating, cleaning and preprocessing was done using Python and supporting libraries such as numpy, pandas, scikit-image, opencv and sqlite. As the dataset is large, loading of features from file systems would consume more time. To reduce the execution time of the similarity matching algorithm, we extract image features only once and store them in a Sqlite database. The Python API connects to the database to retrieve all records and run the algorithm.

3.2 Data visualization

This section explains in detail the rationale behind every visualization, the reasons for data and chart selection and the color scheme used.

3.2.1 Distribution map (North America)



Scientific name: Opuntia x columbiana

Genus: Opuntia

Location: Umatilla County, Oregon, USA

Habitat: Sandy soil

Figure 2: Summary of cacti distribution over North America

Data - The common and most obvious feature that can be extracted from this type of data is the geographic location. And this summarizes the fact that our data only reflects cacti in North America.
Second step was to identify related data attributes that are co-related with the geo location and qualify as more details in association. And so, the subset of attributes used for this chart include, scientific name, genus, location (county, state and country) and the habitat describing the latitude and longitudes pinned in the graph. As part of details the tooltip displays the image of cactus found at this location with the details mentioned above.

Chart - The goal of this chart was to summarize the location limits of underlying data and distribution of cactus across the region. A map of North America seemed like the obvious choice to achieve this goal. We decided to eliminate border details on zoom as we wanted to focus on North America as a region. With this visualization one can clearly identify the region where cacti are distributed across the continent. One would expect the distribution to be limited across the deserts of Arizona and Nevada, however, as the chart clearly represents; it goes beyond this geography.

Visual aesthetics: The color blue naturally depicts water and was the reason for this choice. For land we use a lighter shade of grey as the land is not the most important aspect of this map, but the location pins are. To set an appropriate amount of contrast so that the color of land does not overshadow these pins, we decided to choose two colors on a diverging scale. On one end (black) we use a lighter shade to assign less focus and the darkest shade (red) for the highlights of the graph [12].

3.2.2 **Choropleth map**

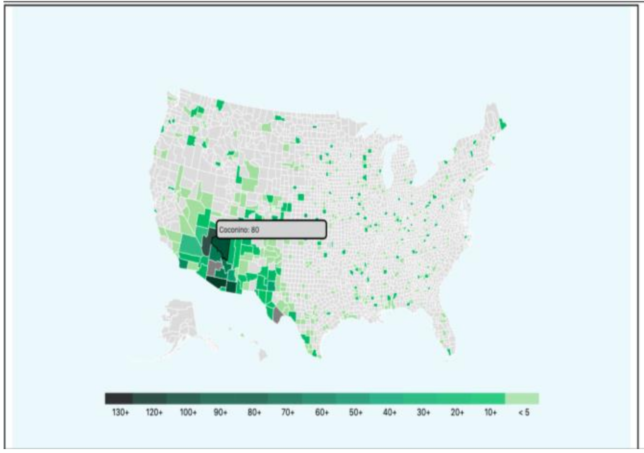


Figure 3: Overview of cacti density throughout the United States by county

Data - Firstly, we filter out the counties of the United States from our dataset. Following that, we compute the number species that are found in each of the counties. Then we create a subset of the dataset containing county name and total number of occurrences of all species. Additionally, we also need the geo coordinates of the counties to display which were obtained in GeoJson format [11].

Chart - As a follow-up to the previous visualization, this choropleth map visualizes how the density distribution of cacti varies across the United States. Looking at this chart, one can evidently say that the cactus vegetation is concentrated in the south eastern region and then decreases as we move radially outwards. The main reason for this is the dry and arid climate of the region which is best suited for the growth of the cacti.

Visual aesthetics - Single hue progressions or color scales are used to show the variation of the density in that region. These progressions fade from a dark shade of green color to a very light or white shade of relatively the same hue. The darkest represents the region with highest density of cactus while the lightest region represents the lowest density. The grey area represents regions with no cactus occurrences. The color green has been chosen because it represents the color of nature.

3.2.3 **Time series graph**

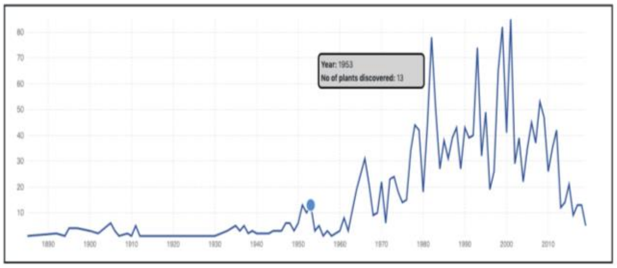


Figure 4: Time series aggregation of number of cacti species identified each year

Data - The data taken into consideration is the year and number of cactus species discovered and added to the dataset for that year. This data was aggregated separately from the original set.

Chart - For representing this information we have chosen a line chart. The data corresponds to change in number over a period of time and line charts are generally used to capture the essence of a time series. This is a line chart where X-axis represents the year and Y-axis represents the number of cactus plants discovered. This chart helps us to understand the period where there were maximum discoveries. The period from 1982 - 2001 showed a significant amount of discoveries.

Visual aesthetics - On hovering over the line a tooltip appears with the Year and no of discoveries made. A blue color is used because this color represents a neutral view. Y-axis ranges over the minimum and maximum number of plants identified throughout. The gridlines make it easier to locate data for a particular year. They have been dimmed so as to not overpower the actual information to be represented.

3.2.4 Network map

Data - To implement this chart we required two attributes which are Scientific name of the cacti and Genus name to which it belongs to.

Chart - The goal of this chart is to create a cluster of cacti species which belong to the same genus. This would help users recognize the genera that contain more species of cacti and what their scientific name are. We can easily recognize that Opuntia is the most common genus type followed by Echinocereus.

Visual aesthetics - In this chart the center notes of the clusters represent the genus of the cluster. These center nodes are connected to other nodes which represent the cacti species which belong to that genus. The genus node or the center node is colored royal blue so as to distinguish it from the nodes.

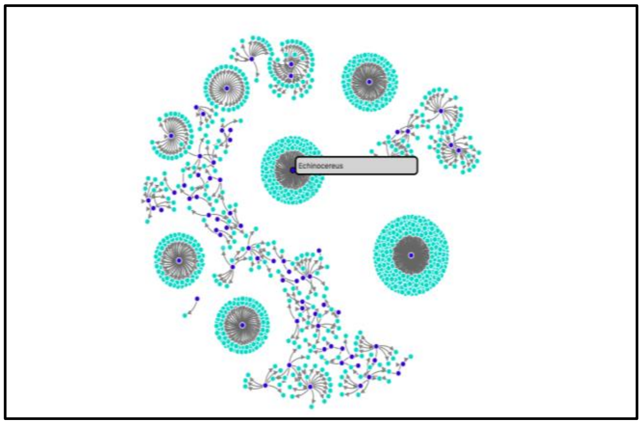


Figure 5: Clusters of cacti grouped by specific genus

3.2.5 Tree map

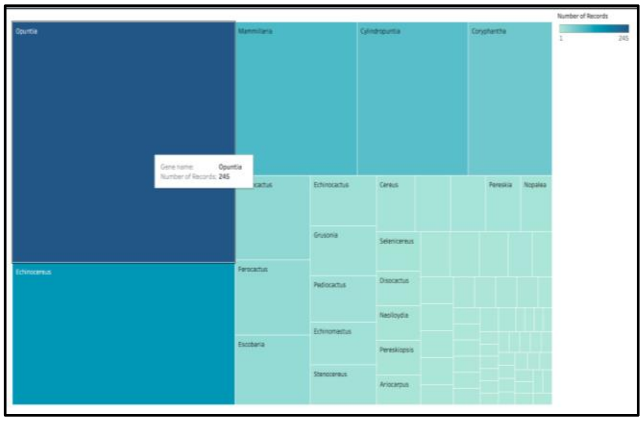


Figure 6: Tree map to identify most common species of cacti genus

Data - This chart gives numerical information about the genera of cacti present in the dataset. It provides a slightly different perspective than the network visualization. It uses the genus name attribute of the dataset. We then compute the number of occurrences of that genus in the dataset and create a subset of the dataset.

Chart - The goal of the chart is to visualize the number of occurrences of the genera in the dataset. In this chart, an area assigned to each genus whose size is proportional to the number of species belonging to that genus.

Visual aesthetics - Single hue progressions are used to show the variation in the number of species in the genera. As the number of species increases, the size of the rectangle increases and the color of the rectangle becomes darker. On hovering over the rectangle one can see the exact number of species belonging to that genus.

3.2.6 Bubble chart

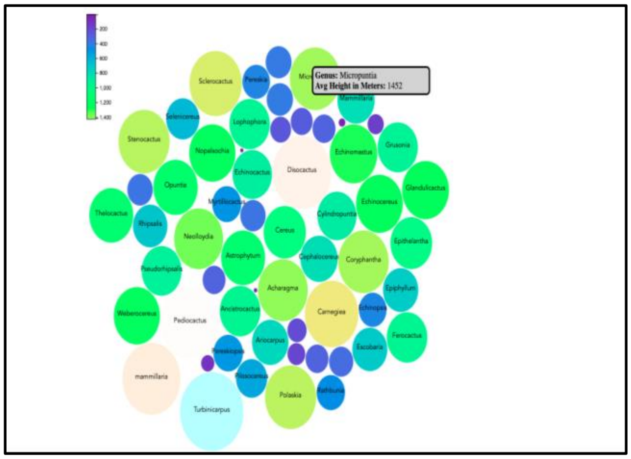


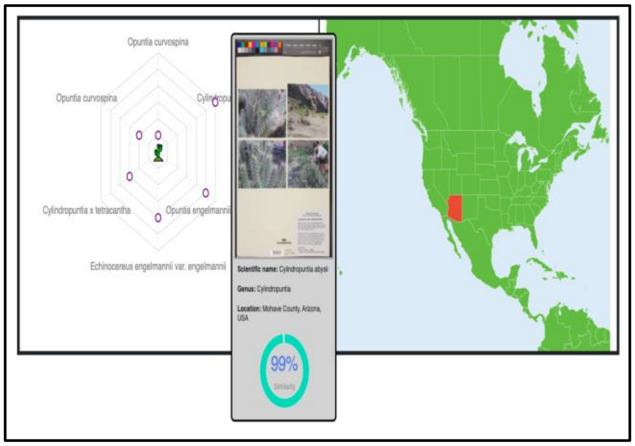
Figure 7: Different Genus with their average elevation from sea level

Data - Scientists are interested in exploring different trends and patterns after looking at the data. Minimum elevation at which a plant is present is one of the important features to be explored. Primary motivation of the chart is to tell the users about different genus of the Cactus family and at what height from the sea level it can be found. This insight is very useful for the user as it will determine where the cactus can be found near the sea level or on mountains at greater heights.

Chart - The graph chosen to represent this data is a bubble chart. Where different bubbles represent different types of Genus in the Cactus Family. Here with the help of the size and color of the bubble we can easily identify the Minimum Elevation of the Genus from the sea level. On hovering over the bubbles, a tooltip appears with the Genus name and height in meters. Legend represents the association between color and height.

Visual aesthetics - The color scheme used for this chart is “cool”. The motivation behind choosing this color scheme was to represent that darker the blue it is nearer to sea level and greener the color represents that it is farther from sea. The size of the bubble also further helps in representing the Minimum elevation. Smaller the size of the bubble it is nearer to the sea level.

3.2.7 **Radar chart with map for displaying similar cactus plants**



Scientific name: Cylindropuntia abysii

Genus: Cylindropuntia

Location: Mohave County, Arizona, USA

Figure 8: Display six similar cactus plants to the uploaded cactus image along with their location on the map.

Data - This part internally uses the similarity matching algorithm mentioned in section 3.1.3. The input can be any image file stored on the local machine.

Chart - This visualization is the key feature of our application. Users may upload any image to get suggestions regarding the species present in the image. Upon upload the file path will be sent to the backend api which will load the image and perform feature extraction on it. Using the extracted features, the algorithm will compute chi-squared distances between the input and previously stored data. Once computation is done, six images with least distance to the input will be returned back to the frontend in the form a json response. The order of images in the response will maintain the ranking obtained at the backend.

For representation we use a radar chart, as they are primarily used to show commonality and measurements in ordinal data. As we have used a distance metric to calculate similarity, we consider images to be an ordinal input. Each element of the graph corresponds to ‘better’ in some respect [10]. And so, the radial nature of the graph provides a good basis to show relative distances and ranks of images. There are 6 levels in the chart. Each level corresponds to the rank of images. The image with least distance shall be placed in the innermost level, to give the impression of closeness. Subsequent images will occupy levels outward.

The algorithm can be generalized to return k nearest neighbors where k is any arbitrary number. Various experiments have shown that symmetric figures are more appealing to the human mind [11]. To incorporate this design principle, we use a symmetric shape, hexagon, for visualizing the radar chart. With this the value of k was fixed to 6.

This tool is built for anyone who is interested in exploring the flora around them. However, as the intelligence of the system has limitations. To present a better set of results, we associated location with the image, represented by the map on the right side of the screen. On hovering over a result, a viewer gets to see more details for the species such as name, genus and location. Most importantly, the tooltip shows the similarity score for that particular image. The state in which the species is found is then highlighted on the map. Users may use this functionality to discard any result not applicable based on their current location or the location of the plant in their input.

Visual aesthetics - The levels of the radar chart are placed merely to provide reference of distance from the input. The input can be imagined to be placed at the center of the graph. And so, we use a lighter shade for representing the radar chart. A color with enough contrast with the background is used to mark the position for each result. The color was chosen such that the overall theme did not turn monotonous. Similarly, for the tooltip, we use a light color such that it is clearly visible against the white background.

The map color scheme is kept simple. Naturally land is associated with green or brown, however the general shade of the cacti images in the result is brown, so we choose green. Water is naturally blue. And as discussed before, red is popularly used to focus on important things, in this case it being the state of the plant in consideration. Green and red are opposites on a diverging color scale and hence best serve the purpose.

4 METHODOLOGY

Our visualization caters to two types of audience:

1. Plant enthusiasts
2. Scientists

We have chosen our data visualization, intelligence, attributes and features keeping in mind the needs of these users and answered the questions which they are likely to have while analyzing the data.

Our application answers the following questions with the help of different interactive graphs.

Motivation for choosing Similarity as our Intelligence Quotient: Let us consider a situation where a person is strolling in a garden and finds a cactus that interests them. They decide to

explore more about similar cactus plants, for example, what other similar plants exist, where they are located and how they are related to each other, etc. Also, scientists are curious to understand the different trends and patterns in the data, but it is difficult to understand it from the raw information present on the SEINet website. In this case, our application will help them understand the merits of the data.
As engineers, we too struggled to see the strength of this data. And so, after looking at different attributes and their relations we hope that people are likely to find it easier to develop use cases for this dataset.

Our visualization will help such enthusiasts answer the following questions by simply uploading an image.

a. Find 6 Similar Cactus Plants to the Cactus Image Uploaded by the user.
b. Find the Location of those Similar Cactus plants
c. Find the Genus of those plants and understand which genus Look similar to each other.

Questions answered with the help of our Visualizations which will help Scientists analyze the data with more precision and will give insights which have true importance in determining answers to their queries are:

1. Find where different types of genus are found with respect to elevation from sea level. This information helps understand the growth pattern of different kinds of cacti families and could be used to identify regions for further exploration.

3. Display the density of different cactus types in North America. This provides answers to biodiversity rich areas which must be preserved.

4. Understand the trend of discoveries of cacti plants and gather information about periods during which more new plants were identified. This may raise questions regarding how new species are formed and whether slower discovery rates indicate loss of biodiversity and environmental degradation.

5. Which cacti belong to which family and which ones are rare and hard to find.

5 EVALUATION PLAN

In this project, we have built an intelligent tool to analyze the similarity between various plants. As a user centric model, we have included various types of features.

We implemented two texture descriptor algorithms and two distance metrics in order to understand which combination of algorithm and distance metrics gives a better accuracy. After analyzing manually, the output, we concluded that Local Binary Pattern when combined with Chi-Squared Distance gives more accurate image similarity for this set of images. Because HoG considers only edges and one direction of each pixel rather than all the 8 directions in a pixel, LBP also calculates local patterns carefully which further helps us in deciding that LBP descriptor will help us in getting a more accurate and similar image. Chi-squared distance is more preferred as compared to Euclidean because of its innate feature of supporting histogram data as output of LBP and HoG algorithms is a combination of Histograms of each cell.

Another feature which we evaluated was that in North American land the distribution and density of Cactus Plants is more in Arizona and Mexico. An interesting observation that we were able to make from our product was that in few areas where the density of cactus was high, they had similar type of plant families near to each other. We analyzed this by taking the network chart and the density of cacti based on the elevation from sea level. For instance, as Arizona is elevated far off the sea level, we can find more numbers of species in this area which are linked to specific genus values.

6 DISCUSSION AND FUTURE WORK

The aim of our project is to analyze the SEINet CACTACEAE dataset and present it in a visually appealing manner. With the help of our analysis, we were not only able to provide the details of cacti in a visual form; but also, we were successful in finding some trends between different attributes. However, this system has its limitations. The image matching algorithm currently focuses solely on the texture obtained from images. The algorithm can be modified to consider categorical attributes such as genus, location and habitat to improve the accuracy of the algorithm. The user input can be extended to allow the user to provide additional information that will help identify the neighbors better.

REFERENCES

[1] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distribution”
Pattern Recognition, vol. 29, no.1, pp. 51–59, 1996.

[2] N. Dalal, B. Triggs, "Histograms of oriented gradients for humandetection in Computer Vision and Pattern Recognition", 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886893, 2005.

[3] J. Yu, Z. Qin, T. Wan, X. Zhang, "Feature integration analysis of bag-of-features model for image retrieval", Neurocomputing, vol. 120, pp.355-364, 2013.

[4] Adrian Rosebrock (December 7, 2015) Local Bi-nary Patterns with Python OpenCV. Retrieved from: https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/ .

[5] Leila Kabbai ; Aymen Azaza ; Mehrez Abdellaoui ; Ali Douik. "Image matching based on LBP and SIFT descriptor".2015 IEEE 12th International Multi-Conference on Systems, Signals Devices (SSD15).

[6]Scholarpedia.http://www.scholarpedia.org/article/Local_Binary_Patterns
[7] Eric.clst.org.https://eric.clst.org/tech/usgeojson/
[8] LearnOpenCV. https://www.learnopencv.com/histogram-of-oriented-gradients/
[9] BlogGraphIQ. https://blog.graphiq.com/finding-the-right-color-palettes-for-data-visualizations-fcd4e707a283
[10] Wikipedia. https://en.wikipedia.org/wiki/Radar_chart
[11] Michael Bauerly ; Yili Liu.
"Effects of Symmetry and Number of Compositional Elements on Interface and Design Aesthetics"
doi.org/10.1080/10447310801920508
[12] Colorbrewer.
https://colorbrewer2.org/#type=diverging&scheme=RdGy&n=8
[13] Wikipedia.
https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
[14] Wikipedia.
https://en.wikipedia.org/wiki/Local_binary_patterns