

# CUSTOM CHART ANALYSIS

## Motivation:

Leveraging Generative AI, we are working to create a product that significantly enhances the reliability of reading and interpreting charts. LLMs often struggle with analyzing charts and extracting information, particularly when the data isn't directly visible in the image. Our development efforts are focused on ensuring accuracy and efficiency in extracting and analyzing chart data. Here's an overview of our current progress and the steps we've taken so far.

## Poor Chart Reading Performance:

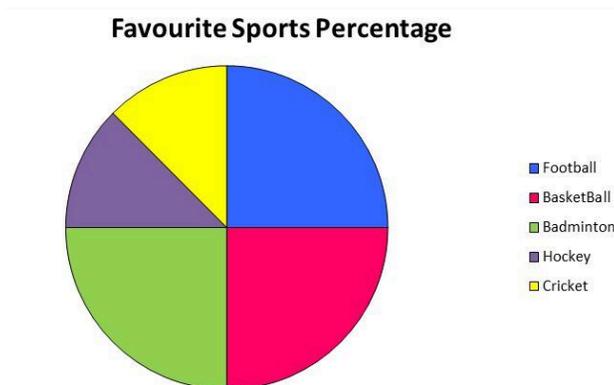
We observed significant errors when testing the Anthropic model and OpenAI vision model on bar charts and pie charts. These tests were conducted using a variety of [bar charts](#) and [pie charts](#). You can view a small sample by following the links above. The models often failed to perform manual calculations accurately, causing it to frequently misinterpret values. For instance, the model assigned the same value to clearly different sections of pie charts, and consistently under- or overestimated values in bar charts; making the results unreliable for precise data extraction and analysis.

We conducted a test in the DiligentIQ staging front end, where we used the models directly without any additional software to aid their performance. This allowed us to evaluate each model's capabilities in a raw, unassisted state.

The following examples show how the Anthropic and OpenAI vision models failed to accurately analyze bar and pie charts during our recent test. The chart images used during testing were sourced from this [Hugging Face Dataset](#) and serve as test cases to evaluate the models' ability to extract and interpret data from visual charts.

1. Pie Charts - Prompt: Report back the pie chart names and corresponding values. As you can see in the model responses below, the numbers are significantly off.

Chart Provided:



## Anthropic Response:

**ANTHROPIC** claude-3-5-sonnet-20240620

1. Charts, Graphs, and Diagrams: The shape is a circle and it is divided into five colored sections. This is a pie chart showing the percentage distribution of favorite sports. The colors used are blue, pink, light green, purple, and yellow.

2. HTML Data Representation:

Sport	Percentage	Color
Football	30%	Blue
Basketball	25%	Pink
Badminton	25%	Light Green
Hockey	12.5%	Purple
Cricket	7.5%	Yellow

3. Organizational and Flow Charts: Not applicable to this image.

4. Color Differentiation: The pie chart uses distinct colors for each sport. Football is represented by a bright blue, Basketball by a vibrant pink, Badminton by a light green, Hockey by a muted purple, and Cricket by a bright yellow.

5. Logos: There are no logos present in this image.

6. Maps: This image does not contain any maps.

7. Other Elements:

- The image has a title at the top that reads "Favourite Sports Percentage" in bold black text.
- There is a legend on the right side of the pie chart, listing the sports and their corresponding colors.
- The image has a thin blue border around its edges.

## Open AI Response:

**OPENAI** gpt-4o The image contains a pie chart titled "Favourite Sports Percentage." The chart is divided into five segments, each representing a different sport. The segments are color-coded and labeled in a legend to the right of the chart. Below is a detailed analysis and HTML data representation of the chart.

**Detailed Analysis:**

1. Charts, Graphs, and Diagrams:

- The shape is a circle (pie chart) and the colors are:

- Blue: Football
- Red: Basketball
- Green: Badminton
- Purple: Hockey
- Yellow: Cricket

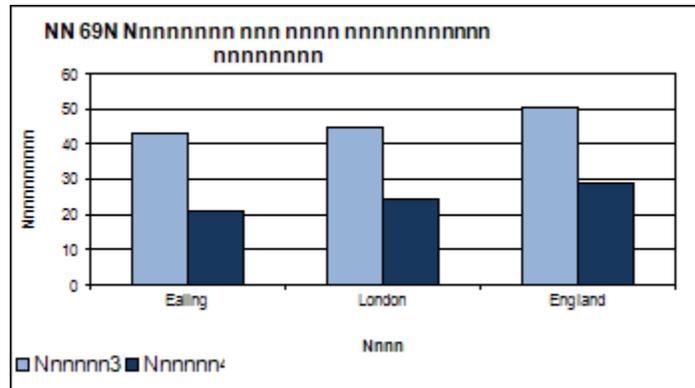
- The chart is divided into five segments, each representing a different sport.

2. HTML Data Representation: html

Sport	Color	Percentage (Estimate)
Football	Blue	20%
Basketball	Red	20%
Badminton	Green	20%
Hockey	Purple	20%
Cricket	Yellow	20%

## 2. Bar Charts - Prompt: Please report back the bar chart names and corresponding values.

### Chart Provided:



## Anthropic Response:

**ANTHROPIC** claude-3-5-sonnet-20240620 The image contains a bar chart comparing three locations: Beijing, London, and England. The shape of each bar is a rectangle and the color is light blue for the higher bars and dark blue for the lower bars.

The chart shows two data points for each location, represented by the two bars of different heights. The y-axis is labeled "Percentage (%)" and ranges from 0 to 80 in increments of 20.

Here's an HTML table representation of the data:

Location	Light Blue Bar (%)	Dark Blue Bar (%)
Beijing	55	45
London	58	48
England	62	50

The title of the chart appears to be partially cut off or obscured, making it difficult to read the full context. The visible part reads "NN 69N Nnnnnnnn nnn nnnn nnnnnnnnnnnn nnnnnnnnnn".

At the bottom of the image, there's a partially visible legend or note that reads "Nnnnnn3 N Nnnnnn".

The chart uses a simple design with a white background and black grid lines. The x-axis labels (Beijing, London, England) are clearly visible below each pair of bars.

## OpenAI Response:

**OPENAI** gpt-4o The image contains a bar chart with three groups of bars representing data for three different locations: Ealing, London, and England. Each group has two bars, one in light blue and one in dark blue. The chart includes a legend, axis labels, and a title.

1. Charts, Graphs, and Diagrams:

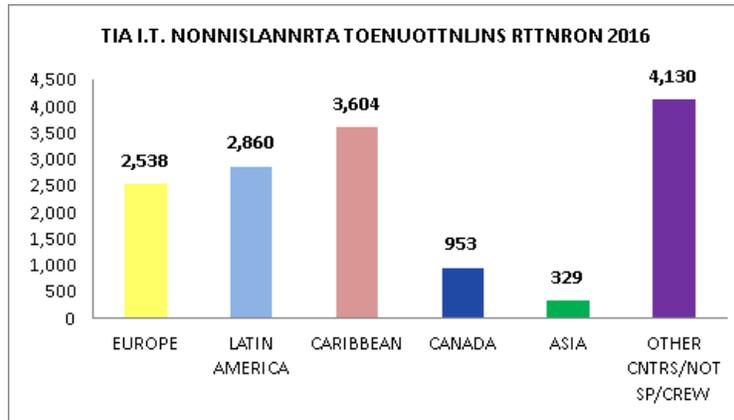
- The shape is a rectangle and the color is light blue. This bar represents "Nnnnnn3" data.
- The shape is a rectangle and the color is dark blue. This bar represents "Nnnnnn4" data.
- The x-axis is labeled "Nnnn" and includes the categories: Ealing, London, and England.
- The y-axis is labeled "Nnnnnnnn" and ranges from 0 to 60.
- The title of the chart is "NN 69N Nnnnnnnn nnn nnnn nnnnnnnnnnnnnn nnnnnnnnnn".

2. HTML Data Representation: html

Location	Nnnnnn3 (Light Blue)	Nnnnnn4 (Dark Blue)
Ealing	40	20
London	50	30
England	55	35

We observed that the LLMs performed extremely poorly when required to analyze charts and perform calculations. However, the LLMs performed adequately when the data was present directly in the image and no calculations were needed. As you can see in the example below, both Anthropic and OpenAI are adequate at extracting data directly from the image.

Chart Provided:



Anthropic Response:

OpenAI response:

ANTHROPIC claude-3-5-sonnet-20240620

1. Charts, Graphs, and Diagrams: The image contains a bar chart. The shape of each bar is a rectangle. The colors of the bars are as follows:

- Europe: Yellow
- Latin America: Light blue
- Caribbean: Pink
- Canada: Dark blue
- Asia: Green
- Other CNTRS/NOT SP/CREW: Purple

The chart has a vertical axis showing values from 0 to 4,500, and a horizontal axis listing different regions.

2. HTML Data Representation:

Region	Value	Color
Europe	2,538	Yellow
Latin America	2,860	Light blue
Caribbean	3,604	Pink
Canada	953	Dark blue
Asia	329	Green
Other CNTRS/NOT SP/CREW	4,130	Purple

OPENAI gpt-4o

### Image Analysis

#### 1. Charts, Graphs, and Diagrams

- **Bar Chart:**

- **Europe:**

- The shape is a rectangle and the color is yellow.
- The height of the bar corresponds to a value of 2,538.

- **Latin America:**

- The shape is a rectangle and the color is light blue.
- The height of the bar corresponds to a value of 2,860.

- **Caribbean:**

- The shape is a rectangle and the color is pink.
- The height of the bar corresponds to a value of 3,604.

- **Canada:**

- The shape is a rectangle and the color is dark blue.
- The height of the bar corresponds to a value of 953.

- **Asia:**

- The shape is a rectangle and the color is green.
- The height of the bar corresponds to a value of 329.

- **Other Cntrs/Not SP/Crew:**

- The shape is a rectangle and the color is purple.
- The height of the bar corresponds to a value of 4,130.

## Initial Chart Analysis Using the LLM:

Using OpenAI's vision model powered by Generative AI, we first identified the type of chart we wanted to extract data from. This test evaluated bar and pie charts — we will include line and area

charts in future testing. We then queried the LLM to determine if data was present within the chart. If no data was detected, we performed manual analysis using the code developed at DiligentIQ. Manual analysis involves thoroughly reviewing and interpreting the chart data by applying custom algorithms to ensure accuracy.

Before proceeding with manual analysis, we query the LLM for the significant RGB values corresponding to labels in the charts. The LLM performs moderately well in identifying these values. To improve accuracy, we use a cosine similarity maximization algorithm, which helps to match the colors and labels more precisely. In simpler terms, this algorithm compares the RGB color values provided by the LLM with the actual colors in the chart, finding the closest match by measuring how similar they are. This process reduces the chances of errors in identifying the correct labels for each color in the chart.

## Manual Analysis Main Packages:

- **AWS Textract:** Utilized with a custom adapter for enhanced text extraction capabilities.
- **OpenCV:** Open Source Computer Vision Library used for advanced image processing.
- **Scikit-learn (Sklearn):** Employed for machine learning and statistical modeling for data analysis and clustering.

## Pie Chart Manual Analysis:

We began by using OpenCV's Hough Circle Transform function to identify the location of the pie chart. The Hough Circle Transform is a technique used to detect circles in images, which helps us accurately locate the pie chart within the image. Documentation for OpenCV's Hough Circle Transform function can be found [here](#).

With the pie chart isolated, we then applied color quantization to address subtle yet significant differences in the RGB values of each pie chart section. This process simplifies the color palette of the image, making it easier to analyze. We achieved this by using the k-means clustering algorithm from Sklearn. In layman's terms, k-means clustering groups similar colors together, reducing the complexity of the image. Documentation for the k-means clustering algorithm can be found [here](#).

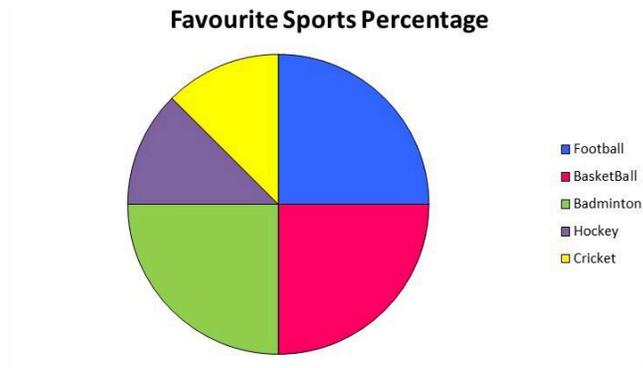
Next, we used the RGB values provided by the LLM and the cluster centers found from the k-means clustering algorithm. We matched labels to actual RGB values using cosine similarity, a measure of similarity between vectors. Cosine similarity compares the angle between two vectors (in this case, color values) to determine how similar they are. Each possible pair was tested, and the combination that yielded the highest similarity was used to match colors to labels. Simply put, we found the closest color match for each label.

With the colors isolated, we calculated the number of pixels for each color and compared it to the total number of pixels identified by the Hough Circle Transform. This gave us the approximate percentage for each color, which we then reported along with its corresponding label. Simply put, we counted the pixels of each color and calculated what fraction of the pie chart they represent.

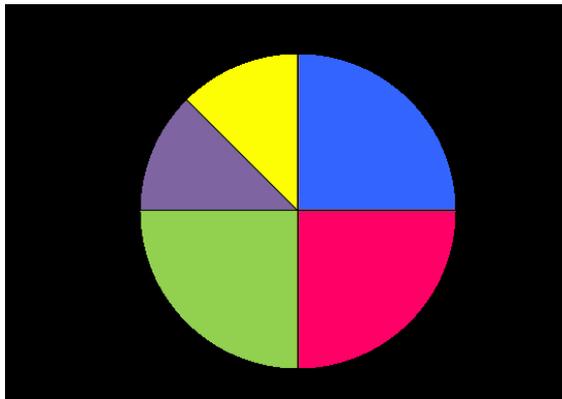
This process ensures that even minor differences in RGB values are accurately accounted for, leading to precise label matching and percentage calculation for each section of the pie chart. By combining advanced image processing techniques with machine learning, we can accurately interpret the data in pie charts, ensuring reliable results.

Compared to OpenAI's and Claude's performance, our approach shows a significant increase in accuracy, demonstrating the functionality and purpose of this code effectively. See the example below,

Chart Provided:



Isolated Pie Chart after color quantization:



Isolated piece of the pie chart:



Manual Analysis Response:

**CUSTOM** HTML representation of the data extracted from the image:

Cricket	11.967385656386611
Badminton	24.78599088110614
BasketBall	24.98995723134566
Hockey	12.119680531232117
Football	24.569786549852253

Favourite Sports Percentage

## Bar Chart Analysis:

We began by using a custom adapter on AWS Textract to extract information regarding the y-axis. AWS Textract is a service that automatically extracts text and data from scanned documents. In this case, we trained a custom adapter to recognize and extract numbers on the y-axis of our charts. With this information, we were able to determine the y-axis scaling factor, which represents the number of pixels corresponding to changes in value on the y-axis. In simpler terms, the scaling factor helped us understand how the chart's height (in pixels) related to the actual data values.

With the scaling factor identified, we applied preparation filters to the image to reduce noise. Noise reduction involves removing any unwanted distortions or artifacts in the image, making it clearer and easier to analyze.

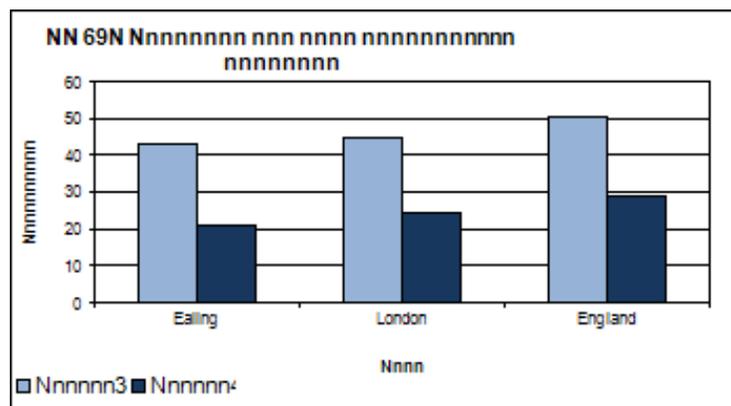
Next, we used color quantization techniques, as described above, to standardize the RGB values. This step simplifies the color palette, ensuring that similar colors are grouped together. We then isolated bar charts of the same color and applied the Canny Edge detection algorithm. Canny Edge detection is a technique used to identify the edges of objects in an image, which in this case helped us outline the bars in the chart. Documentation for Canny Edge detection can be found [here](#).

Using the Canny Edge-detected image, we applied the findContours function, which uses a variant of the Teh-Chin chain approximation algorithm; documentation for this function is available [here](#). The findContours function highlights the edges of objects in the image. By utilizing the Structural Analysis and Shape Descriptors function, documentation [here](#), we were able to estimate the number of edges in the detected contour. If the contour has four edges and meets a minimum height requirement, we can assume it represents a bar. We then calculated its height in pixels and, using the y-axis scaling factor, determined its actual value. In layman's terms, we counted the pixels to find the bar's height and convert it to the actual data value using the scaling factor.

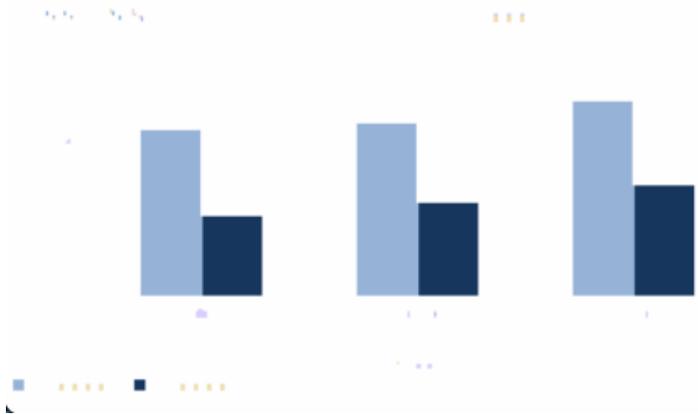
Next, we used AWS Textract to search for words near and below the bar to find its x-axis label, expanding the search region horizontally until a label was found. This step ensured we accurately identified the corresponding label for each bar.

If the bar chart was color-dependent, we used cosine similarity, as described above, to match the bar to its corresponding color label. Cosine similarity helped us match the RGB values of the bar with the predefined color labels. This approach ensures accurate extraction and interpretation of bar chart data, demonstrating the effectiveness and purpose of the code. By combining advanced image processing techniques with machine learning, we can reliably interpret and analyze bar charts, ensuring precise results.

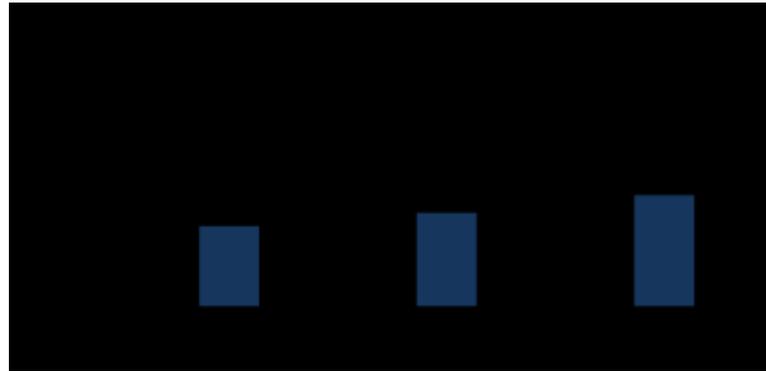
Chart Provided:



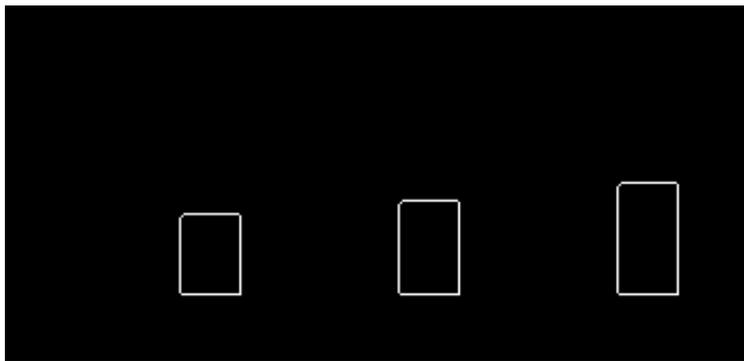
Noise reduction and color quantization:



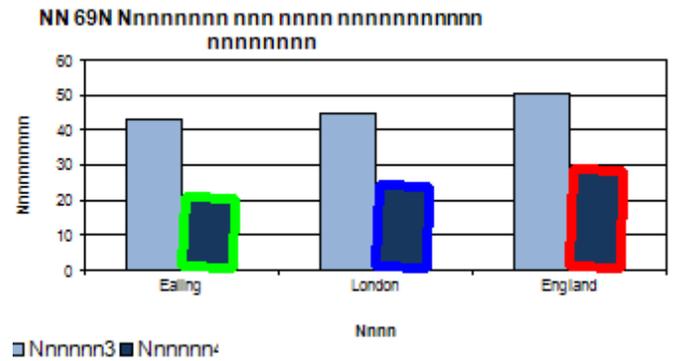
Isolated bars:



Canny Edges:



Original Image with detected bars outlined:



Results:

**CUSTOM HTML representation of the data extracted from the image:**

Ealing NNnnn3	43.28
Ealing NNnnn4	20.47
London NNnnn3	45.04
London NNnnn4	23.98
England NNnnn3	50.88
England NNnnn4	28.66

Compared to OpenAI's and Claude's performance, the DiligentIQ approach shows a significant increase in accuracy, demonstrating the functionality and purpose of this code effectively.

## Failed Solutions

**Pie Charts:** Our first approach involved measuring the circumference of each section, which proved ineffective. We then tried using a range of RGB values to differentiate sections, but this method was also unsuccessful, leading us to implement color quantization to standardize the RGB values. Initially, we attempted a greedy algorithm for cosine similarity, but it did not yield good results, so we switched to maximization for the entire group. We also tried eliminating black, gray, and white pixels to reduce noise but struggled with this approach. Ultimately, we included these colors in the quantization process and disregarded them during analysis.

**Bar Charts:** We first tried using Hough lines to detect bars, but this method resulted in too much noise and misidentification of bars, including axis lines and reference lines. Touching bars were not recognized as separate bars. AWS Textract also performed poorly in recognizing x-axis labels, prompting us to introduce a custom search method.

## Conclusion:

This project aimed to enhance the accuracy and reliability of chart interpretation. By integrating Generative AI with advanced techniques, we significantly improved our model's ability to accurately interpret and analyze chart data, setting new standards for accuracy and reliability. Through rigorous testing, we identified significant limitations in the current capabilities of LLMs, specifically the Anthropic and OpenAI vision models. These limitations include performing accurate identification of bar and pie charts. Initial methods, such as measuring circumferences and using a range of RGB values, proved ineffective. Additionally, the use of a greedy algorithm for cosine similarity and attempts to eliminate specific pixel colors did not yield the desired accuracy.

Our persistence led us to discover more effective techniques, such as applying color quantization and maximizing cosine similarity, which significantly improved results. These methods, combined with advanced image processing and machine learning, allowed us to get our model to accurately interpret and analyze chart data.

This project highlights DiligentIQ's commitment to innovation and precision. While the journey presented numerous challenges, the outcomes reinforce our capability to develop robust and reliable data analysis solutions. By integrating these advanced techniques into our platform, DiligentIQ continues to lead in delivering exceptional data analysis tools for private equity research, setting new standards for accuracy and reliability.