Constellation Detection Using KNN Classifier

Rishabh Verma Shohom De Aryan Kumar Toshika Kamble

July 8, 2024

• **Abstract**

This report presents the methodology and implementation of a constellation detection system using a K-Nearest Neighbors (KNN) classifier. The system aims to classify images of constellations into predefined categories. The approach involves preprocessing images to extract relevant features, training a KNN classifier, and evaluating the model's performance.

1 Introduction

Constellation detection is an essential task in astronomical image processing. This project uses a KNN classifier to identify constellations from images. The system processes images to extract features representing stars and their relative positions, then uses these features to train and evaluate a KNN model.

2 Methodology

2.1 Image Preprocessing

The preprocessing steps include:

- **Loading Images:** Images are loaded from the specified directories.
- **Color Channel Extraction:** The red and blue channels are extracted to highlight stars and constellation lines.
- **Binarization:** Thresholding is applied to convert the image to binary format, isolating stars and lines.
- **Noise Reduction:** Median filtering is applied to reduce noise.
- **Edge Detection:** Canny edge detection is used to find edges in the binary image.
- **Contour Detection:** Contours are detected to identify star positions.

2.2 Feature Extraction

Features are extracted by normalizing the coordinates of the stars and lines. This involves:

- **Normalizing Coordinates:** Coordinates are normalized based on the brightest and second brightest stars.
- **Angle Calculation:** Angles between stars are calculated to maintain geometric consistency.
- **Feature Vector Construction:** A feature vector is constructed from the normalized coordinates and line information.

2.3 Training the KNN Classifier

The KNN classifier is trained using the feature vectors extracted from the template images. The steps include:

- **Data Preparation:** Feature vectors are padded to ensure consistent length and then split into training and testing sets.
- **Model Training:** The KNN classifier is trained with the training dataset.
- **Model Evaluation:** The classifier's accuracy is evaluated using the testing dataset.

3 Datasets

The datasets used in this project consist of images of various constellations. The template images are used to train the model, while a separate set of images is used for testing. Each image undergoes preprocessing and feature extraction to generate the input for the classifier.

4 Algorithm Description

4.1 Distance and Angle Calculations

- **Distance Calculation:** The Euclidean distance between points is calculated using:

$$d(p1, p2) = \sqrt{(p1_x - p2_x)^2 + (p1_y - p2_y)^2}$$

- **Angle Calculation:** The angle between three points is calculated using:

$$\theta = \arccos\left(\frac{d(p0, p1)^2 + d(p0, p2)^2 - d(p1, p2)^2}{2 \cdot d(p0, p1) \cdot d(p0, p2)}\right)$$

4.2 KNN Classifier

The KNN algorithm classifies input data based on the majority vote of its neighbors. For this project, the number of neighbors k is set to 3. The steps include:

- **Training:** The classifier is trained with labeled feature vectors.
- **Prediction:** The classifier predicts the class of a test feature vector by finding the k nearest neighbors and taking a majority vote.

5 Results

The model's accuracy is evaluated using the testing dataset. Despite preprocessing and feature extraction, in accordance with the current implementation the model is not able to classify constellations properly and resulted in an accuracy of 0%. Further refinements and debugging are required to improve the model's performance.

6 Discussion

The low accuracy indicates potential issues in the preprocessing, feature extraction, or classification steps. Future work includes:

- Debugging the preprocessing pipeline to ensure correct feature extraction.
- Experimenting with different image processing techniques to improve feature quality.
- Tuning the parameters of the KNN classifier or exploring other machine learning models.

7 Conclusion

This project provides a foundation for constellation detection using a KNN classifier. While the current accuracy is low, the methodology and framework established here can be further refined to achieve better performance. The project is based on the approaches outlined in [1] and [2].

References

[1] Kartikeya, et al. "Constellation Detection." `https://github.com/Kartikeya18153/Constellation-detection/blob/master/Report.pdf`

[2] Ji, Liu, Wang. "Constellation Detection Using Image Processing." `https://web.stanford.edu/class/ee368/Project_Spring_1415/Reports/Ji_Liu_Wang.pdf`