INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Team : theBoys

GitHub - https://github.com/Digital-Image-Processing-IIITH/project-theboys

Project 40: Peacock eye counting

Mentor TA - Meher Shashwat Nigam

Team : Astitva Gupta 2018101085 (CSE)
Kartik Agarwal 2018102017 (ECE)
Ayan Biswas 2019121009 (CSD)
Priyanshu Madaan 2018101097 (CSE)

Objective

To design a GUI Tool to identify and label peacock feather eyes (ocelli).

- The problem lies in locating and segmenting (loosely) the ocelli from the image.
- The image may be in varying lighting conditions or the peacock in the image may be photographed from varying camera angles.
- We should be able to provide some solution for extreme cases even if we cannot find a general approach for them.

Target Standard

- Tool should run in real-time and GUI response should not be delayed.
- User should be able to adjust configuration in an easy manner.
- User should be able to play around with the options, observe results regardless of knowledge of Digital Image Processing.
- The application should have an adaptive approach with baseline to set default hyper-parameters for the user.
- For cases where an ocelli or a set of ocelli cannot be detected, there should be some alternative solution that the user can turn to.

Tool Overview

Our tool takes input as an Image. It has default parameters which can be adjusted using various sliders. Our focus has been on adding these to an extent that the user can satisfy himself with it's complexity but not feel dull or overloaded. We have 3 channels of output.

- A noisy detected image. Noise helps discern detection related patterns.
- A bare grayscale image of only the detector response.
- A denoised detected image which shows the detector prediction drawn on the target image.

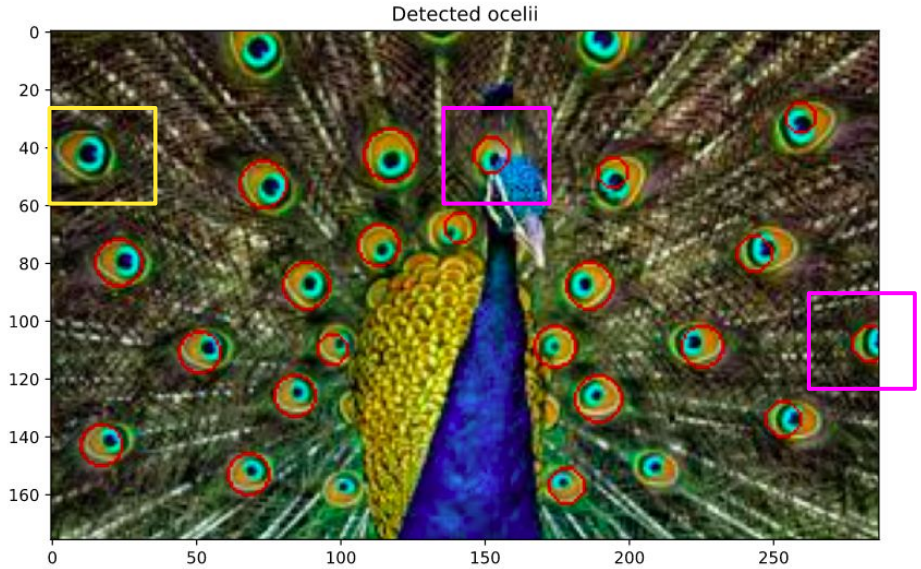
Approach

We have implemented two modes of ocelli detection in our tool. Hough Transform mode and Template matching mode. Hough Transform mode aims to provide a more general solution were as Template Matching mode aims to deal with corner cases by detecting a subset of ocelli based on a template from the image it self.

Hough Transform Mode

- Edge detection is performed on image using Canny detector.
- Elliptical hough transforms were attempted. However, we have settled on Circles due to speed and performance.
- Peaks in Transform domain are inversely mapped to Image domain to obtain circular markers. These identify the Ocelli

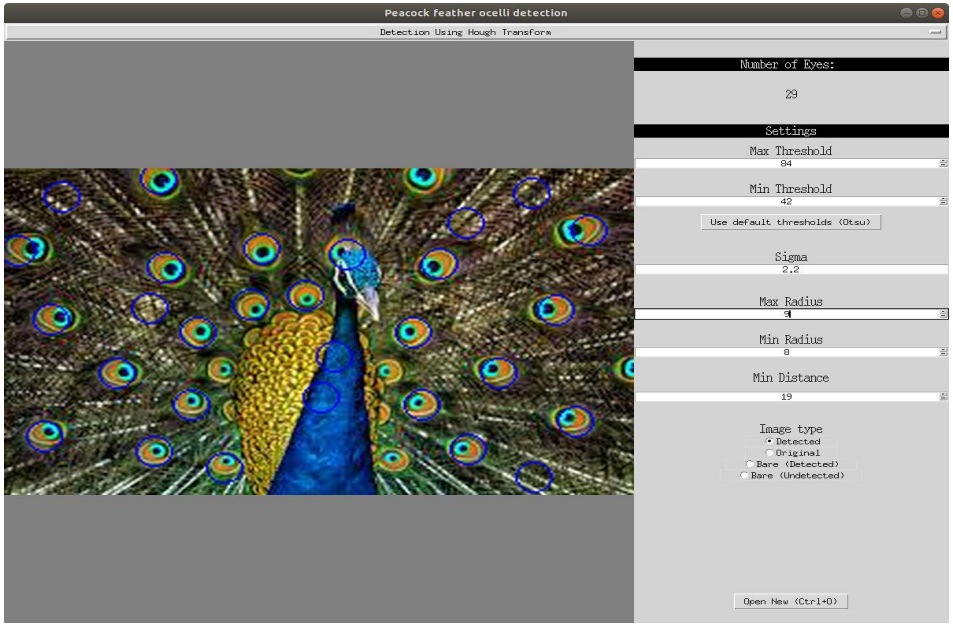
Hough Transform Mode (Cont)



Initial Analysis - Mid Progress

- The ocelli that were detected are marked by Red circles.
- Parameters can be changed further to get more/less ocelli.

Hough Transform Mode (Cont)



Tool View

- Parameters to play with.
- Real Time response
- Great for learning purposes.

Problems

- Some ocelli are elliptical in shape. Our algorithm only detects almost perfectly circular ocelli.
  - ➔ Can be solved using Elliptical Hough transformation which also handles the elliptical edges (but takes a lot of time for detection 😞)
- Canny detector also shows a lot of feathers in the edges which also affect the ocelli and gives ambiguous results sometimes.
  - ➔ Can be solved by taking median/gaussian blur first, then some adaptive thresholding and then ellipse/circle detection.

Template Matching Mode

- A template is selected using the GUI tool. This template can be used to identify the remaining Ocelli.
- We provide 3 matching methods.
- A default template is provided as a placeholder.
- Hope to process some a set of default images such that a raw or transformed input image can use it. This is to somewhat automate the process and reduce need of human assistance.

Template Matching Mode (Cont)

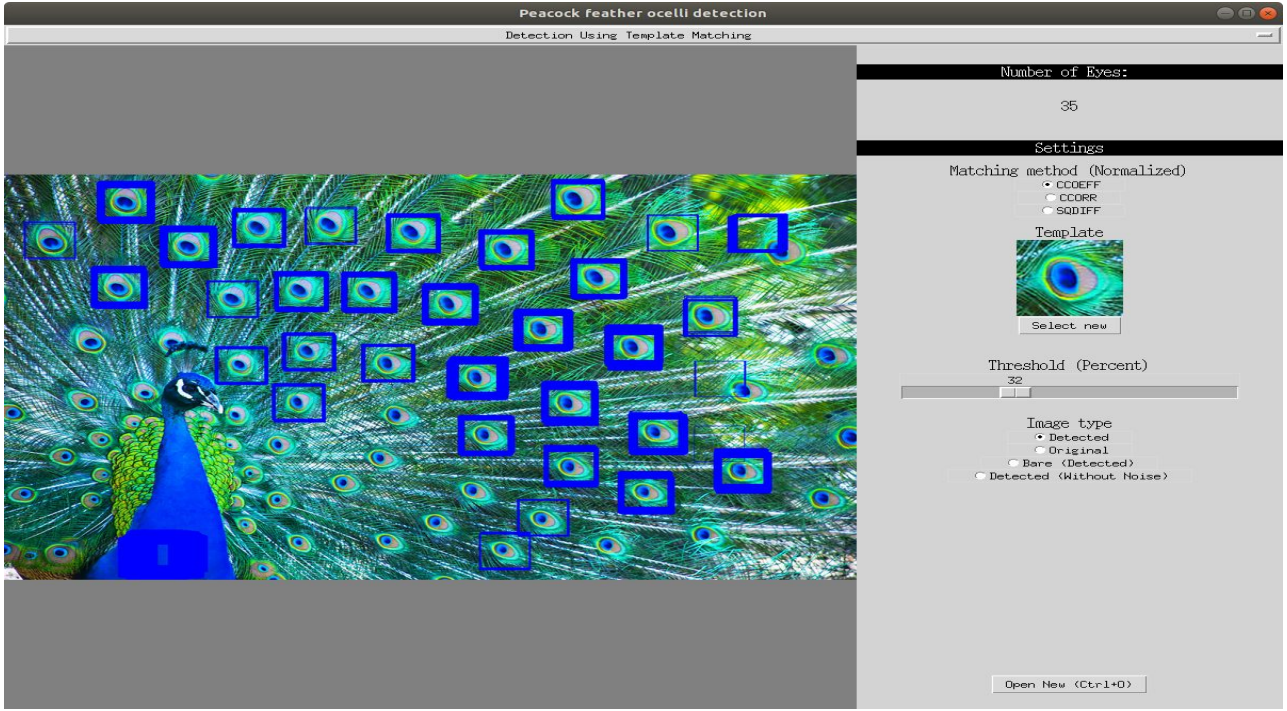
Advantages:

- The difficulty of template selection is invariant to scale and rotation.
- Does not depend on edge detection.

Template Matching Mode (Cont)

- Drastically reduces the need to play around with parameters. Only threshold and a matching method is needed.
- Can be used to select sets of Ocelli that are harder to get via Hough Transform.
- An unintended feature of this mode is to select sets of Ocelli that appear somewhat distinct, as templates would match to similar Ocelli given the threshold. The user can adjust this.

Template Matching Mode (Cont)



This is somewhat difficult to set in Hough Transform domain. Reason for this is difficulty of obtaining a clean edge image. This is also good for an easy selection where settings need not be an headache.

Template Matching Mode: Problems

- Sometimes it may be a hard to adjust the template. By fidgeting the template box slightly about the eyeballed location, the output can be much better.
- This is not rotation and scale invariant. Hence, it tends to fit closer to the input template.
- A solution to this problem is to scale and rotate the selected template, and perform detection using the set of spanned images. This has not been done as of now.
- At low thresholds , the match is satisfied by overlaps which are not structurally similar.

Conclusion:

We are able to detect feathers fairly consistently, and based on Ravi sir's feedback we have made it largely depend on the user as well. This pertains to going beyond the default thresholds for Hough Transform, or selecting a Template. We have done it based on various techniques taught to us in class. To that end, we haven't used statistical/learning approaches. We believe there is quite a bit of work that can be done in the future and have expressed those ideas.

Division of Work

- **Conceptualizing** : Done as a team.
- **Ayan and Astitva** : Notebook preparation, testing and experimenting with validity and proof of concept.
- **Astitva** : Backend for GUI based on concept.
- **Kartik & Priyanshu** : GUI programming, heavy lifting in terms of code.
- **Ayan & Kartik** : Testing and documentation.

Thank You