# Computer Vision 600100 Counting Starfish Student ID: 201601628 Date: **May 19, 2020** Deadline: Tuesday 5th May 2020 by 2pm (+14d)

# 

# Image Processing Pipeline

**Introduction**

The proposed pipeline is a composite of 2, Process1.m and Process2.m. Each is a pipeline within itself. P1 is focused on isolation of starfish blobs without too much concern for mask *quality*, whereas P2 is focused and the quality of masks (as long as all starfish are detected). By feeding the more accurate detections of P1 into P2, the proposed architecture aims to isolate the higher quality masks (i.e. remove erroneous segmentations) produced by P2.

**Exploratory Work**

The design is influenced by extensive exploratory and analytical work, which helped to develop an understanding of the provided data and potentially applicable (and inapplicable) techniques. This, for example, involved the use of MATLAB’s colour thresholding *app,* studying the histograms of images, and research into localised denoising methods. Some examples of this exploratory work are included in the appendix, and associated scripts are included in the project folder, “*. /experimentation*”.

Thresholding

…appropriate colour model

…customisation of export function and determining parameters

Hists

…binarizing a single HSV channel

…then finding it works similarly with RGB single channel

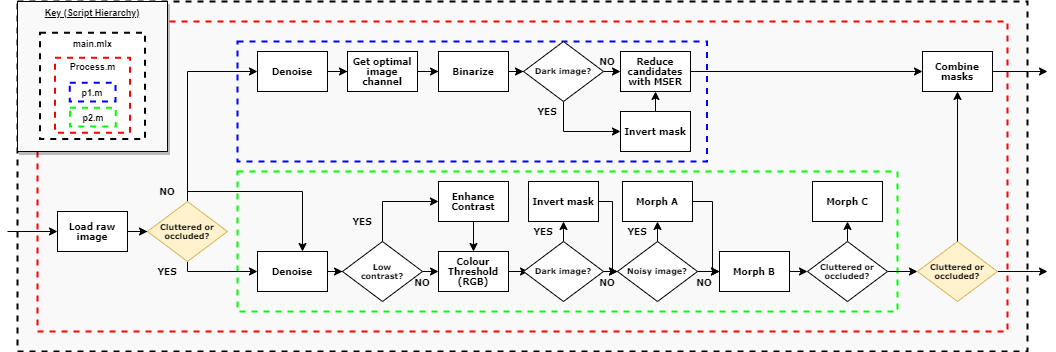
**Proposed Pipeline**

Figure 1: High-level overview of the implemented pipeline architecture (full-size version included in submission).

Steps

P1

-slice select

-binarize

P2

-denoising

-thresholding

-contrast

-dark

-morphing

Give details on each stage of your proposed pipeline.

Explain what is the purpose of each stage in the image processing pipeline.

Explain what algorithm / function is used in each step. Why was it used? What parameters were used and why?

**Testing**

In addition to formative, exploratory work – some lightweight testing of implemented methods is included in the project, in the folder, “*. /testing*”. The goal of this is to validate, demonstrate and justify the techniques employed (e.g. for classifying noise types). A script for processing every given image was also created to aid of these validations.

# Results

Give a figure showing the result of your image processing pipeline on the default starfish.jpg image.

Does your image processing pipeline perform well on the noise and colour variations of Starfish? (show this)

Show the effect of your image processing pipeline on an alternative image (E.g Starfish\_5.jpg) and perhaps if your code works on any other images.

ALL FINAL OUTPUTS (MASK AND ROIS) – maybe go in the appendix

# Discussion

Discuss the results presented in the previous section. What works, and what doesn’t work; including why it may or may not work.

Considering the domain, P1 may be a sufficient pipeline in itself.

P2 has the benefit of actually denoising the original image, should that be useful. It is also better suited to more complex image where the objects to detect aren’t as well isolated.

The combined pipeline aims to find some middle ground between the simplicity of P1 and the additional utility and generalisation of P2.

P1s largest limitations are the

How can the design of your image processing pipeline and code be improved? Are there any alternative functions / algorithms / approaches which may have been more suitable in hindsight. Is there evidence to support this?

CNN, TEXTURE (CORREMAT), EDGE DETECTION (arguably not with noise ims), SURF/FAST?\*. Possibly a similar composite pipeline to detect more advanced images, e.g. detect NOT starfish and combine masks, or combine different colour models etc.

Remove contiguous space (borders) around images

Consider each stage of the image processing pipeline. Consider variations in noise, colourmaps, and image types, including the more challenging images with occlusion and clutter.

# Appendices

* colour thresholding
* all channel hists
* Noise detection & denoising with edge filters
* Textures

…Useless

* + Acquisition
  + Figures

Test Outputs and Additional Images