DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Digital Image Processing

Assignment #02

Student Name: **Muhammad Hamza Tariq – 371070**

Degree/ Syndicate: **43 CE B**

Retinal Image Segmentation and Objects Analysis

Introduction

The aim of this assignment is to use knowledge about spatial enhancement and segmentation to extract optic disc from given images.

Understanding the Assignment

Assignment Overview

The assignment at hand pertains to **Digital Image Processing** focusing on retinal image analysis, specifically targeting the segmentation and analysis of objects within fundus images.

Key Objectives

- **Spatial Enhancement and Filtering:** Employing transformation and filtering techniques for spatial enhancement.
- **Segmentation:** Extracting specific objects, particularly the optic disc, from fundus images.
- **Error Analysis:** Evaluating the accuracy of segmentation through error calculation.

Spatial Enhancement and Filtering

Spatial enhancement techniques involving transformation and image filtering play a vital role in preprocessing fundus images. These techniques aim to enhance specific features and improve the overall quality of the images, thereby facilitating subsequent analysis tasks.

Segmentation of Optic Disc

Understanding Fundus Images

Fundus images, capturing the retina's inner surface, serve as crucial diagnostic tools for various retinal diseases. Key landmarks within these images include the optic disc, blood vessels, and fovea.

Optic Disc Extraction

The primary objective of the assignment is to extract the optic disc from fundus images using spatial enhancement and segmentation techniques. This process involves identifying and isolating the bright region corresponding to the optic disc while mitigating potential false positives from other bright lesions.

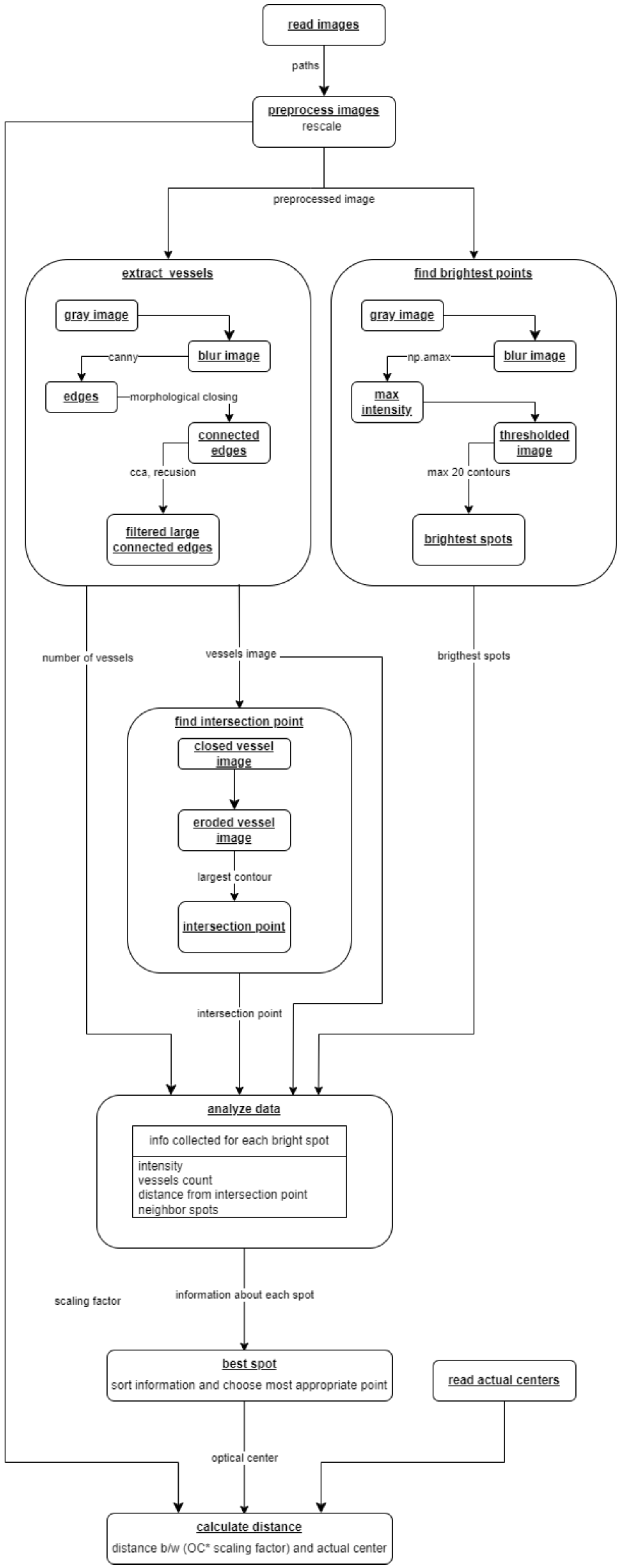
Algorithmic Approach

1. **pre_process_image**:
    - Explanation: This function resizes an image to a specified size and returns the resized image along with the scaling factor used for resizing.

- Working: It calculates the scaling factor required to resize the image to the desired size. Then, it resizes the image using OpenCV's **cv2.resize** function and returns the resized image along with the scaling factor.

2. **extract_vessels**:
   - Explanation: This function extracts blood vessels from a retinal image using image processing techniques.
   - Working: It converts the input image to grayscale, applies Gaussian blur and Canny edge detection to detect edges of blood vessels. Then, it uses connected component analysis to identify connected regions (vessels) and filters out small components based on the minimum component area.
3. **find_brightest_spots**:
   - Explanation: This function finds the brightest spots in an image, typically representing potential locations of the optic disc.
   - Working: It converts the input image to grayscale, applies Gaussian blur, and performs thresholding to identify bright regions. Then, it finds contours of these regions and selects the largest ones as the brightest spots.
4. **find_intersection_point**:
   - Explanation: This function detects the intersection point of blood vessels in an image, which often corresponds to the location of the optic disc.
   - Working: It applies morphological operations to the input image to close gaps between vessel segments. Then, it finds contours and calculates the centroid of the largest contour as the intersection point.
5. **analyze_data**:
   - Explanation: This function analyzes the characteristics of bright spots in an image, including intensity, vessel count, and distance from the intersection point.
   - Working: It iterates over each bright spot, masks the image around the spot, and calculates intensity and vessel count. It also calculates the distance from the intersection point and counts neighboring spots within a certain radius.
6. **find_best_spot**:
   - Explanation: This function identifies the best candidate spot for the optic disc based on vessel count, intensity, distance from the intersection point, and the number of neighboring spots.
   - Working: It sorts the information of each spot based on specified criteria and returns the center of the spot with the highest score, which is considered the best candidate for the optic disc.

Detailed Flow Diagram


read images
paths
preprocess images
rescale
preprocessed image
extract vessels
gray image
blur image
canny
edges
morphological closing
connected edges
cca, recusion
filtered large connected edges
find brightest points
gray image
blur image
np.amax
max intensity
thresholded image
max 20 contours
brightest spots
number of vessels
vessels image
brigthest spots
find intersection point
closed vessel image
eroded vessel image
largest contour
intersection point
intersection point
analyze data
info collected for each bright spot
intensity
vessels count
distance from intersection point
neighbor spots
scaling factor
information about each spot
best spot
sort information and choose most appropriate point
read actual centers
optical center
calculate distance
distance b/w (OC* scaling factor) and actual center

Example information about spot table:

```
Optic Disk Information:
Center          Intensity       Vessel Count    Dst from IP     Nbr Spots
(423, 265)      1.49            1535            23              8
(423, 265)      1.47            1514            28              8
(423, 265)      1.45            1494            33              8
(423, 265)      1.43            1473            26              8
(423, 265)      1.42            1461            40              8
(423, 265)      1.41            1452            14              8
(423, 265)      1.34            1377            17              8
(399, 311)      1.3             1333            52              0
(394, 316)      1.26            1298            59              0
(423, 265)      1.19            1224            32              8
(380, 310)      1.13            1158            62              0
(423, 265)      1.09            1125            39              8
(370, 325)      1.03            1059            80              0
(356, 334)      0.95            976             96              0
(418, 205)      0.83            854             60              0
(330, 360)      0.81            830             133             0
(383, 204)      0.6             612             73              0
---------------------------------------------------
Optic Disk Information:
Center          Intensity       Vessel Count    Dst from IP     Nbr Spots
(320, 229)      1.04            1070            33              9
(320, 229)      1.04            1069            35              9
(320, 229)      1.0             1024            48              9
(320, 229)      0.99            1013            14              9
(320, 229)      0.98            1003            38              9
(320, 229)      0.94            963             22              9
(320, 229)      0.92            947             36              9
(320, 229)      0.92            944             22              9
(320, 229)      0.89            912             28              9
(320, 229)      0.8             819             46              9
(108, 76)       0.42            429             261             0
(57, 189)       0.23            238             266             0
---------------------------------------------------
Optic Disk Information:
Center          Intensity       Vessel Count    Dst from IP     Nbr Spots
(419, 234)      0.63            652             29              5
(419, 234)      0.52            538             26              5
(419, 234)      0.49            505             13              5
(419, 234)      0.49            499             17              5
(419, 234)      0.48            489             9               5
(419, 234)      0.46            476             20              5
(126, 334)      0.15            156             310             0
(248, 493)      0.03            33              310             0
(39, 157)       0.02            22              388             0
(35, 164)       0.01            13              390             0
(33, 169)       0.01            8               391             0
---------------------------------------------------
```

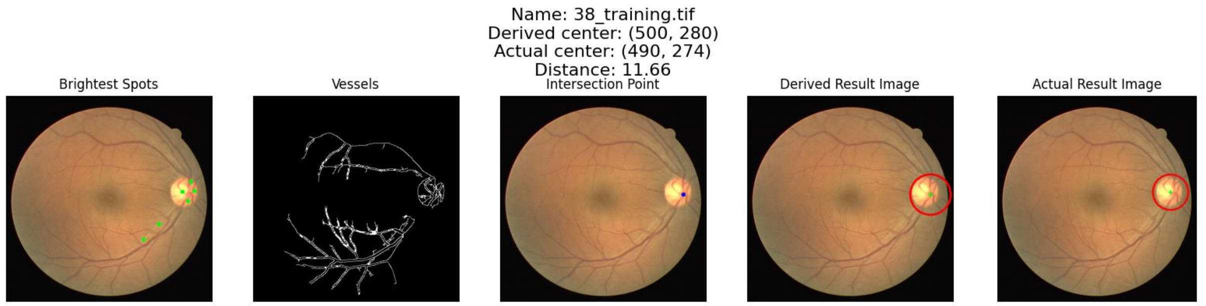
Results:

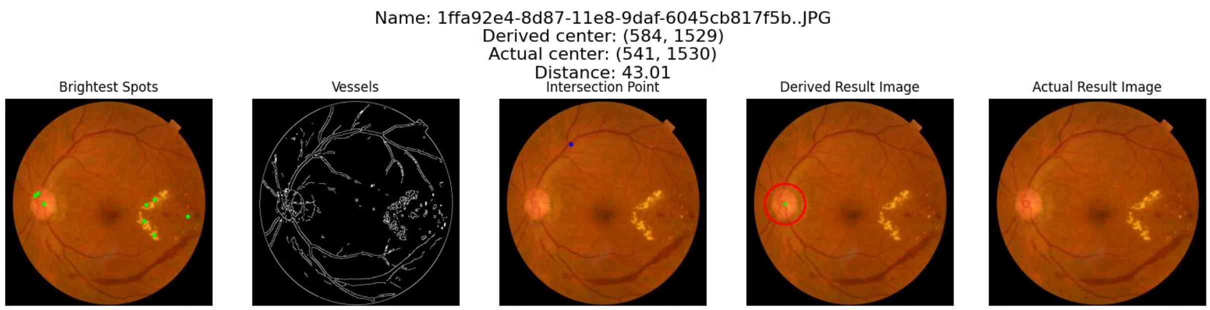
Sr no.	Image	Derived OC	Actual OC	Distance
1	02_test.tif	(466, 302)	(458, 275)	28.16
2	04_test.tif	(353, 261)	(361, 275)	16.12
3	06_test.tif	(462, 266)	(461, 268)	2.24
4	08_test.tif	(455, 278)	(485, 277)	30.02
5	10_left.jpeg	(2208, 2023)	(2439, 1697)	399.55

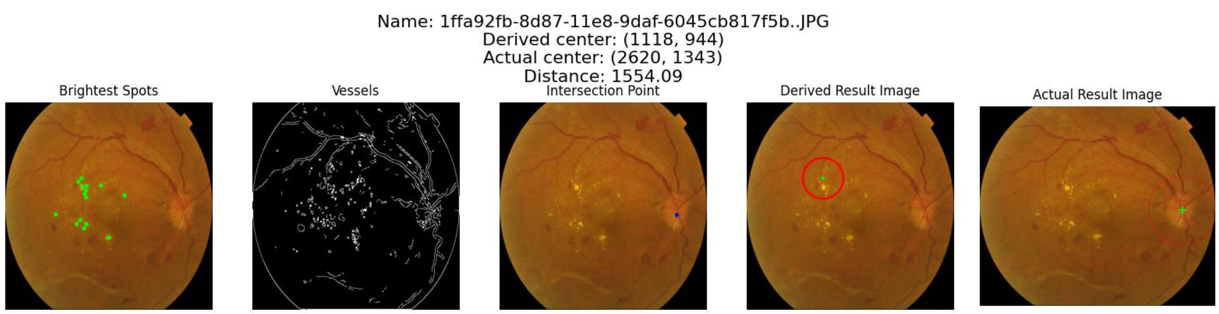
6	10_right.jpeg	(3331, 2320)	(2985, 1562)	833.23
7	10_test.tif	(472, 277)	(468, 278)	4.12
8	12_test.tif	(94, 233)	(82, 257)	26.83
9	13_left.jpeg	(486, 960)	(504, 832)	129.26
10	13_right.jpeg	(2080, 842)	(2119, 852)	40.26
11	13_test.tif	(472, 273)	(486, 268)	14.87
12	14_test.tif	(423, 449)	(479, 275)	182.79
13	15_test.tif	(201, 318)	(193, 282)	36.88
14	16_test.tif	(477, 295)	(479, 258)	37.05
15	17_left.jpeg	(1450, 1042)	(1396, 1322)	285.16
16	17_test.tif	(476, 279)	(467, 267)	15
17	18_test.tif	(451, 277)	(471, 262)	25
18	19_left.jpeg	(1427, 1306)	(1445, 1176)	131.24
19	19_right.jpeg	(2376, 1280)	(2456, 1269)	80.75
20	19_test.tif	(483, 281)	(486, 275)	6.71
21	1ffa92e4-8d87-11e8-9daf-6045cb817f5b..JPG	(584, 1529)	(541, 1530)	43.01
22	1ffa92fb-8d87-11e8-9daf-6045cb817f5b..JPG	(1118, 944)	(2620, 1343)	1554.09
23	1ffa9309-8d87-11e8-9daf-6045cb817f5b..JPG	(1219, 1039)	(609, 1167)	623.28
24	1ffa9523-8d87-11e8-9daf-6045cb817f5b..JPG	(684, 1085)	(652, 1070)	35.34
25	1ffa952f-8d87-11e8-9daf-6045cb817f5b..JPG	(2546, 1255)	(2523, 1246)	24.7
26	1ffa9555-8d87-11e8-9daf-6045cb817f5b..JPG	(452, 1436)	(512, 1501)	88.46
27	20_left.jpeg	(2513, 1508)	(2539, 1513)	26.48
28	20_right.jpeg	(1306, 1285)	(1293, 1296)	17.03
29	20_test.tif	(482, 280)	(482, 284)	4
30	21_left.jpeg	(1327, 2493)	(1724, 1589)	987.33
31	21_right.jpeg	(3146, 1404)	(3036, 1556)	187.63
32	21_training.tif	(173, 163)	(77, 257)	134.36
33	22_left.jpeg	(810, 821)	(747, 848)	68.54
34	22_right.jpeg	(1855, 873)	(1821, 910)	50.25
35	22_training.tif	(494, 282)	(471, 272)	25.08
36	23_left.jpeg	(1306, 1078)	(1417, 1127)	121.33
37	23_right.jpeg	(2073, 597)	(2676, 1242)	882.97
38	23_training.tif	(343, 344)	(428, 227)	144.62
39	24_training.tif	(484, 257)	(472, 289)	34.18
40	25_left.jpeg	(2543, 1442)	(2635, 1455)	92.91
41	25_right.jpeg	(896, 1817)	(791, 1846)	108.93
42	25_training.tif	(466, 278)	(464, 270)	8.25
43	26_training.tif	(101, 236)	(80, 245)	22.85
44	27_training.tif	(483, 287)	(492, 283)	9.85

45	29_training.tif	(496, 293)	(497, 271)	22.02
46	30_training.tif	(470, 292)	(492, 291)	22.02
47	31_left.jpeg	(3214, 2932)	(1665, 1542)	2081.23
48	31_right.jpeg	(3715, 1402)	(3366, 1536)	373.84
49	31_training.tif	(399, 268)	(393, 250)	18.97
50	32_training.tif	(472, 288)	(491, 285)	19.24
51	33_left.jpeg	(2512, 2639)	(1657, 1592)	1351.75
52	33_training.tif	(460, 292)	(470, 305)	16.4
53	34_training.tif	(273, 281)	(388, 223)	128.8
54	35_training.tif	(101, 272)	(81, 276)	20.4
55	37_training.tif	(499, 288)	(496, 292)	5
56	38_training.tif	(500, 280)	(490, 274)	11.66

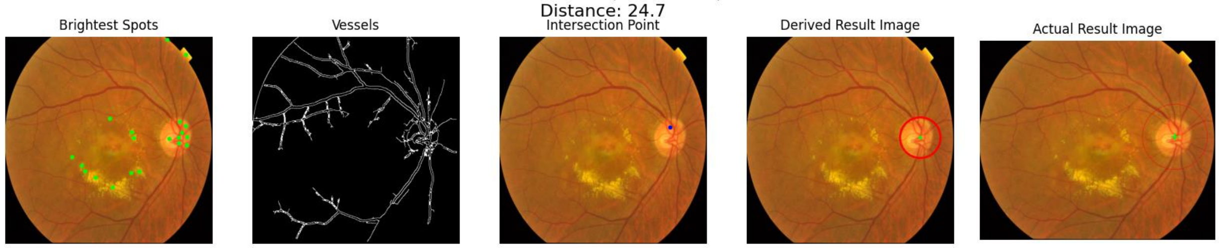
Outputs:



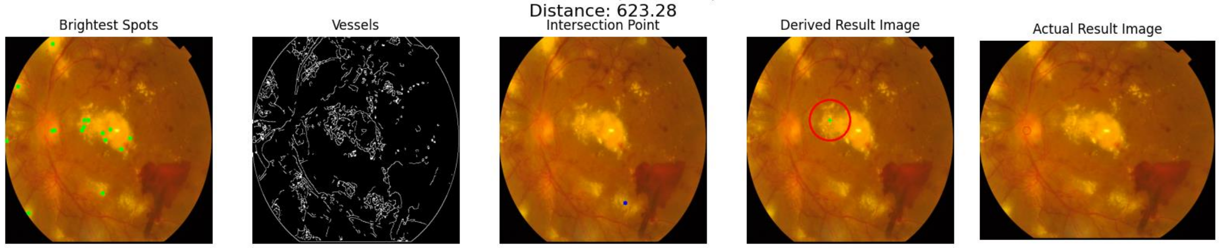




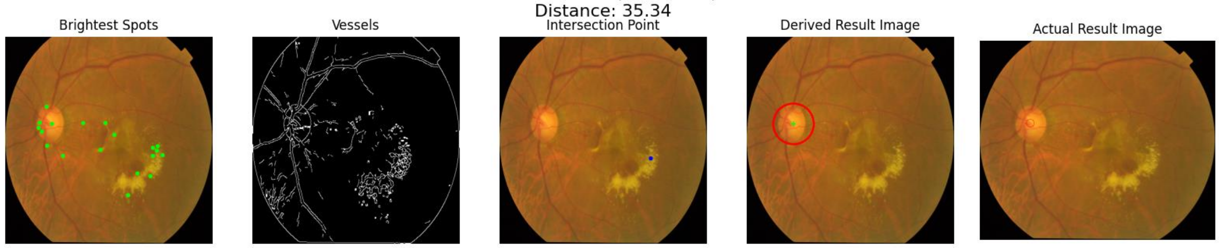
Name: 1ffa952f-8d87-11e8-9daf-6045cb817f5b..JPG
Derived center: (2546, 1255)
Actual center: (2523, 1246)
Distance: 24.7



Name: 1ffa9309-8d87-11e8-9daf-6045cb817f5b..JPG
Derived center: (1219, 1039)
Actual center: (609, 1167)
Distance: 623.28

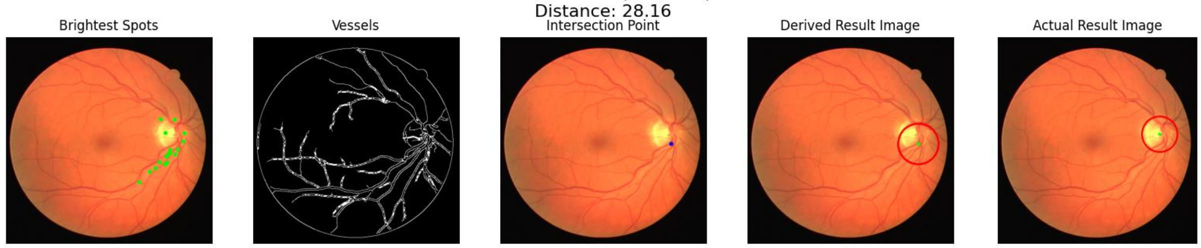


Name: 1ffa9523-8d87-11e8-9daf-6045cb817f5b..JPG
Derived center: (684, 1085)
Actual center: (652, 1070)
Distance: 35.34

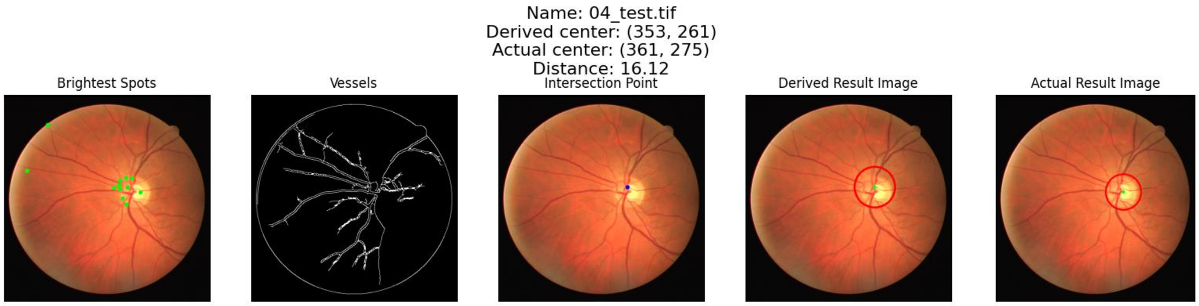


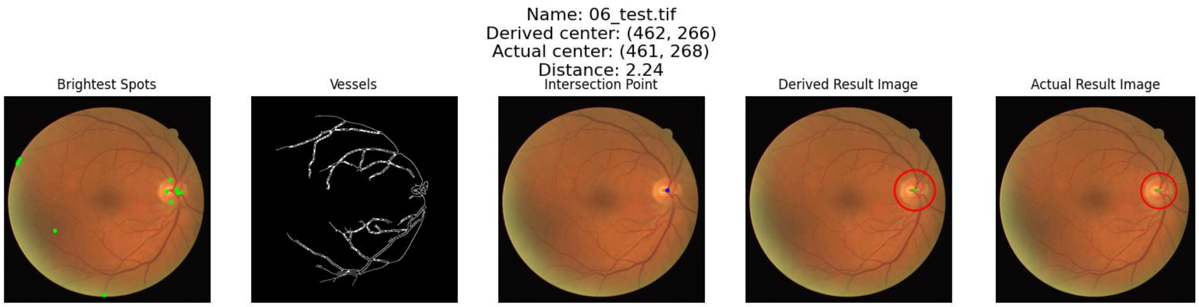
Name: 1ffa9555-8d87-11e8-9daf-6045cb817f5b..JPG
Derived center: (452, 1436)
Actual center: (512, 1501)
Distance: 88.46

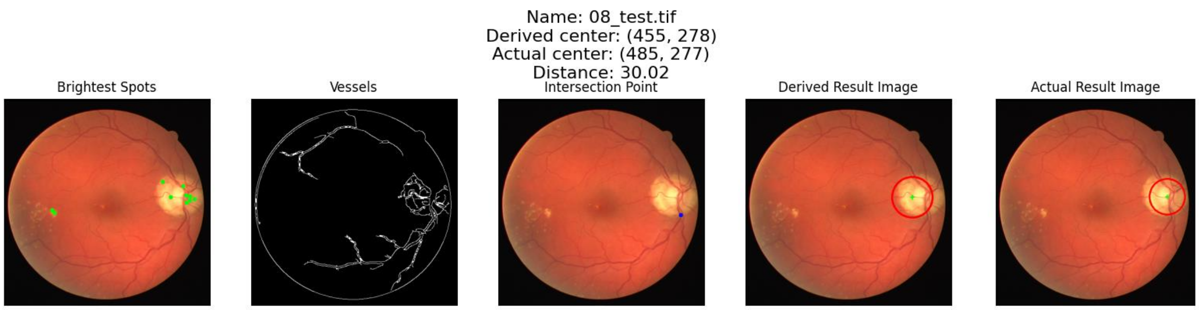
Name: 02_test.tif
Derived center: (466, 302)
Actual center: (458, 275)
Distance: 28.16

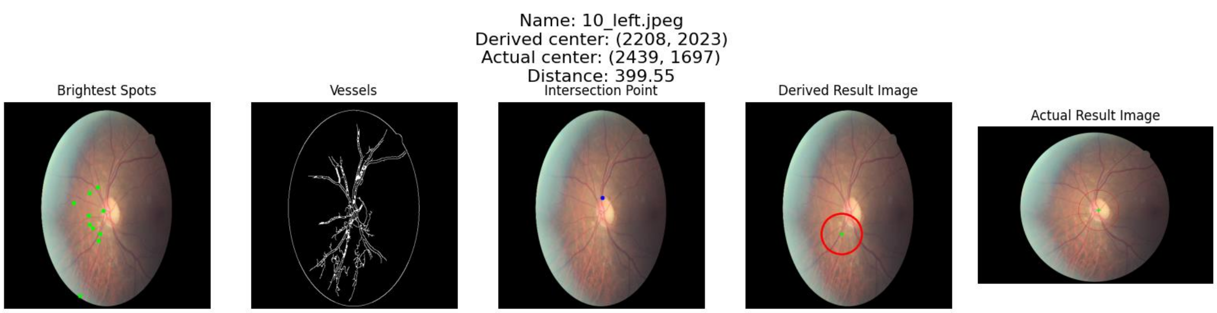


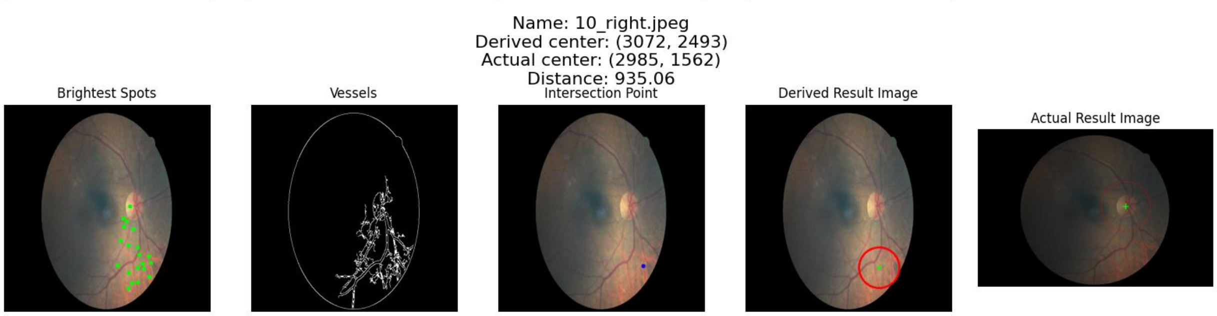
Hamza Tariq 371070



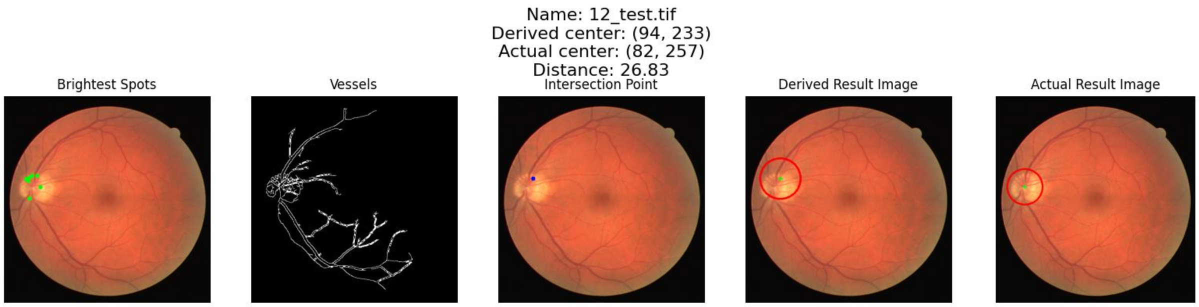


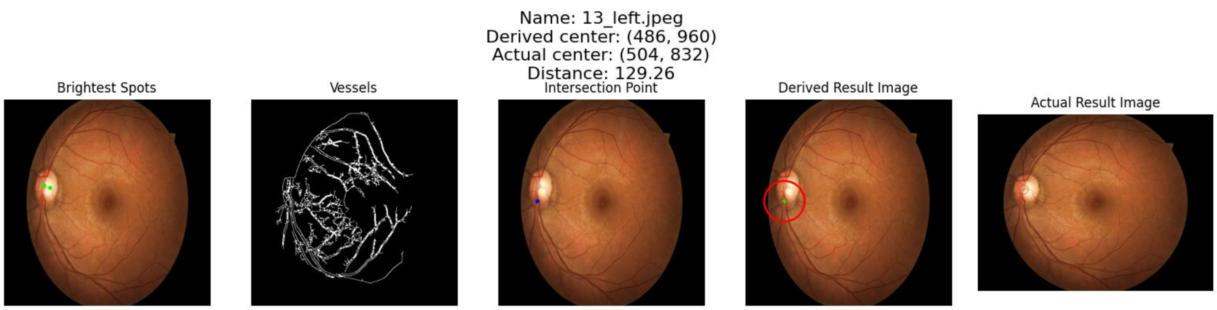


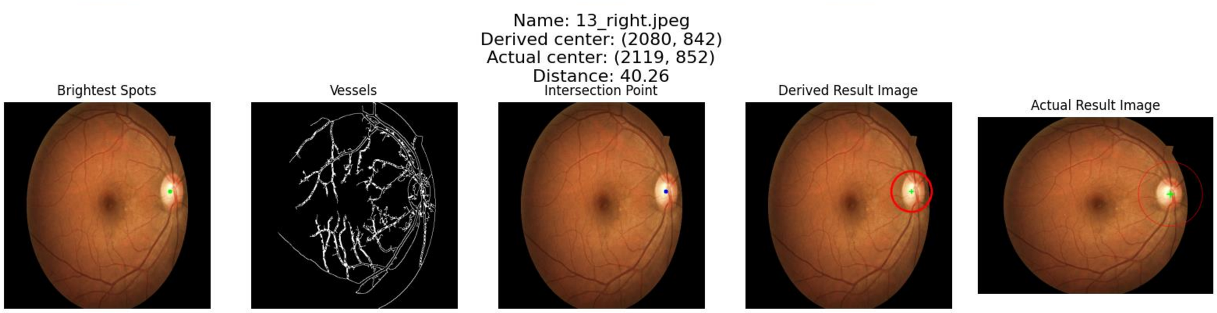


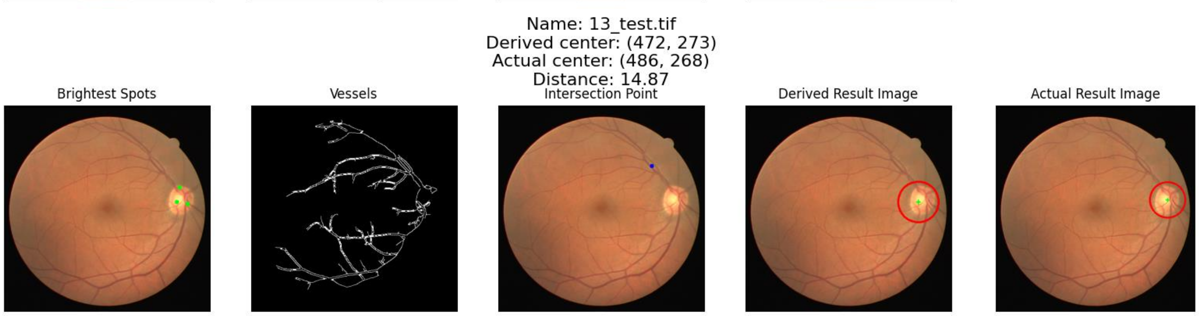


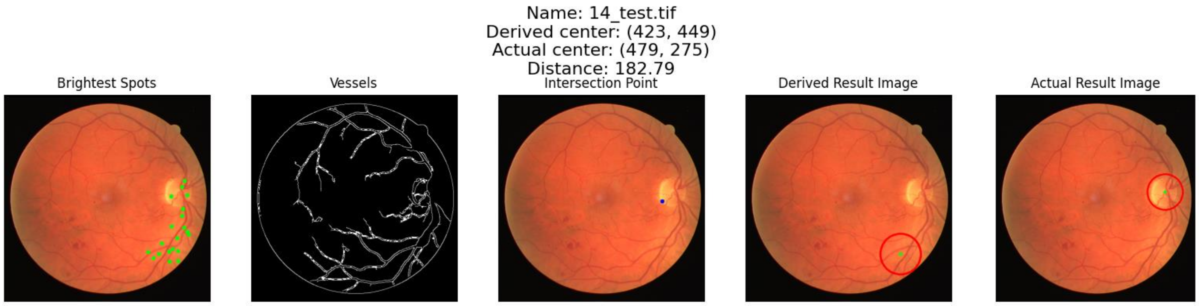

Name: 10_test.tif
Derived center: (472, 277)
Actual center: (468, 278)
Distance: 4.12
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

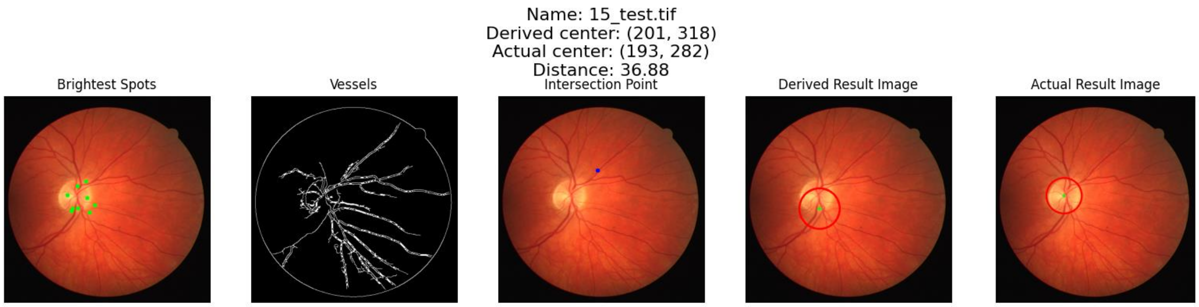

Name: 12_test.tif
Derived center: (94, 233)
Actual center: (82, 257)
Distance: 26.83
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

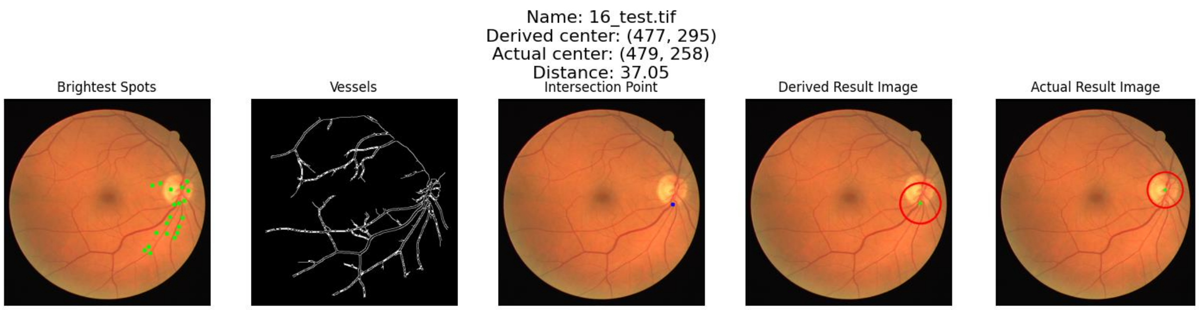

Name: 13_left.jpeg
Derived center: (486, 960)
Actual center: (504, 832)
Distance: 129.26
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

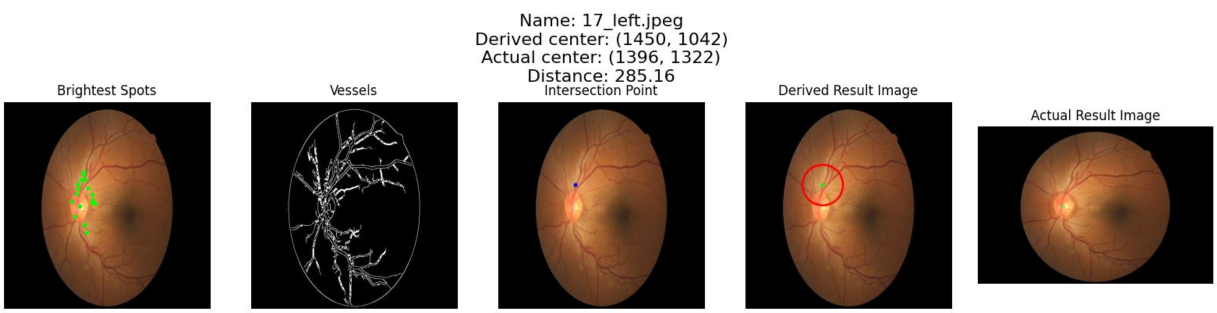

Name: 13_right.jpeg
Derived center: (2080, 842)
Actual center: (2119, 852)
Distance: 40.26
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

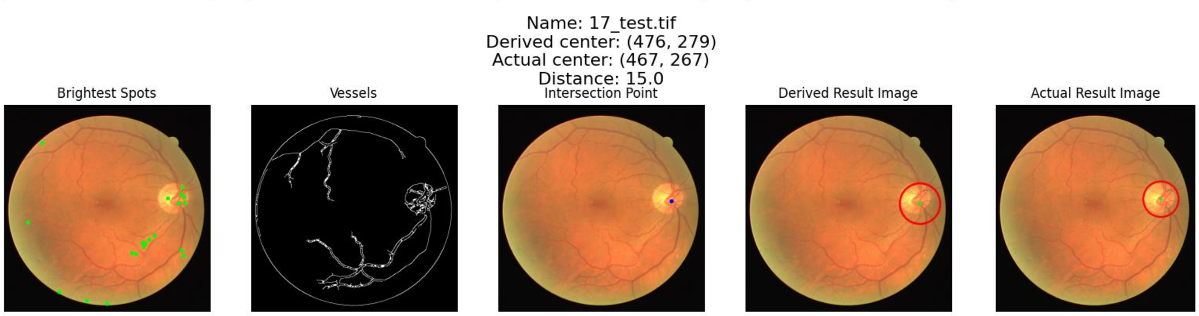

Name: 13_test.tif
Derived center: (472, 273)
Actual center: (486, 268)
Distance: 14.87
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 14_test.tif
Derived center: (423, 449)
Actual center: (479, 275)
Distance: 182.79
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 15_test.tif
Derived center: (201, 318)
Actual center: (193, 282)
Distance: 36.88
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

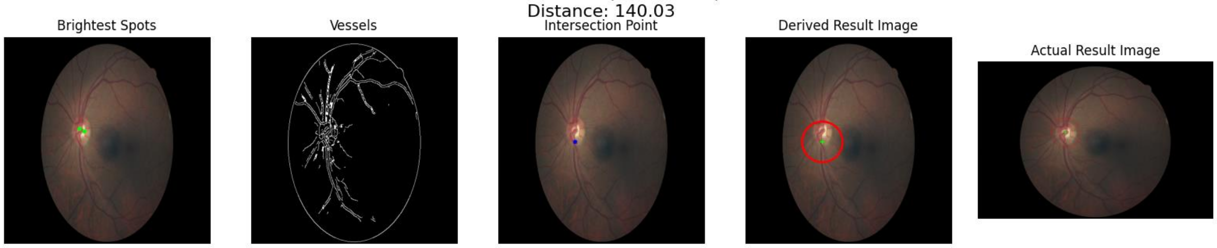

Name: 16_test.tif
Derived center: (477, 295)
Actual center: (479, 258)
Distance: 37.05
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 17_left.jpeg
Derived center: (1450, 1042)
Actual center: (1396, 1322)
Distance: 285.16
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

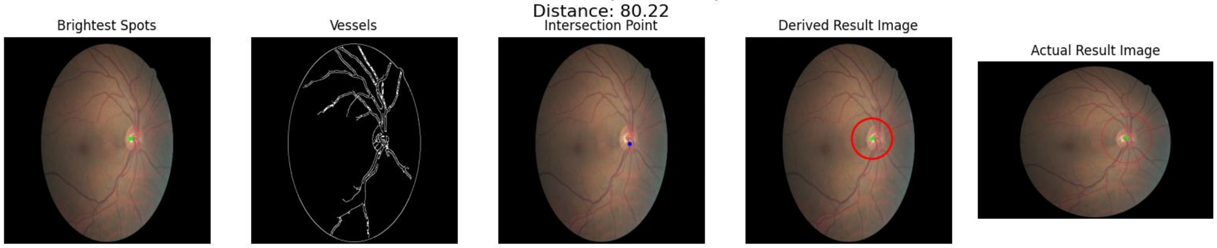

Name: 17_test.tif
Derived center: (476, 279)
Actual center: (467, 267)
Distance: 15.0
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

Name: 18_test.tif
Derived center: (451, 277)
Actual center: (471, 262)
Distance: 25.0

Name: 19_left.jpeg
Derived center: (1442, 1316)
Actual center: (1445, 1176)
Distance: 140.03

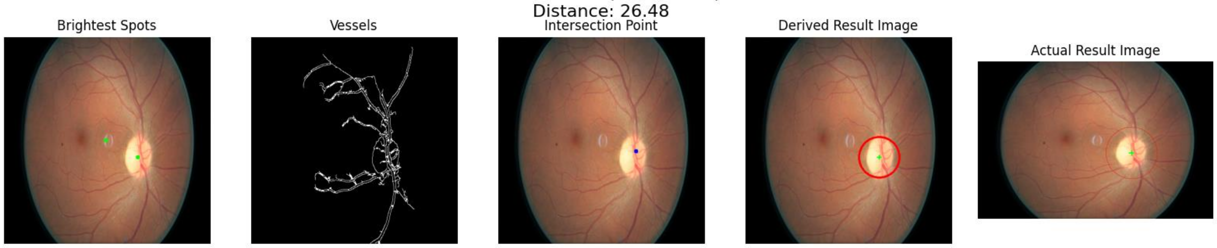


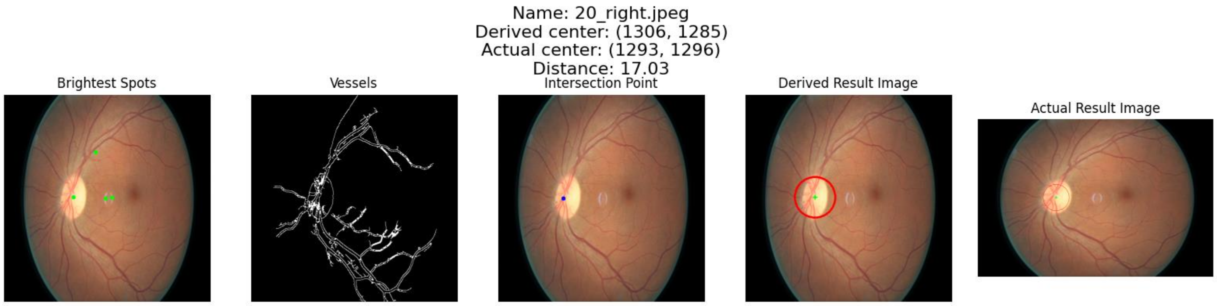
Name: 19_right.jpeg
Derived center: (2376, 1275)
Actual center: (2456, 1269)
Distance: 80.22

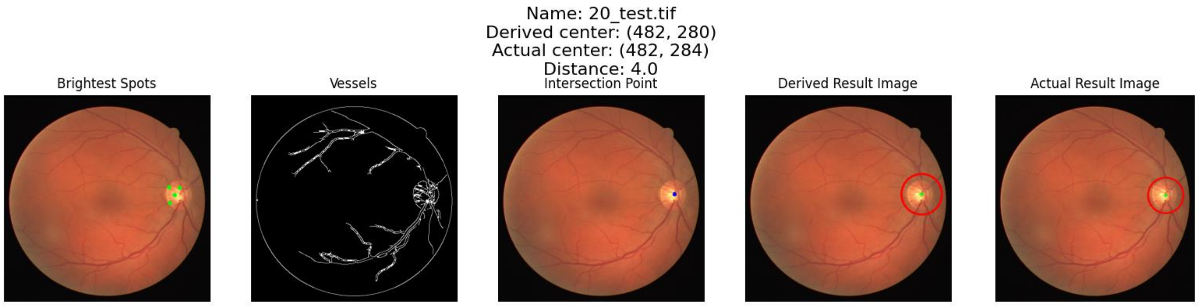


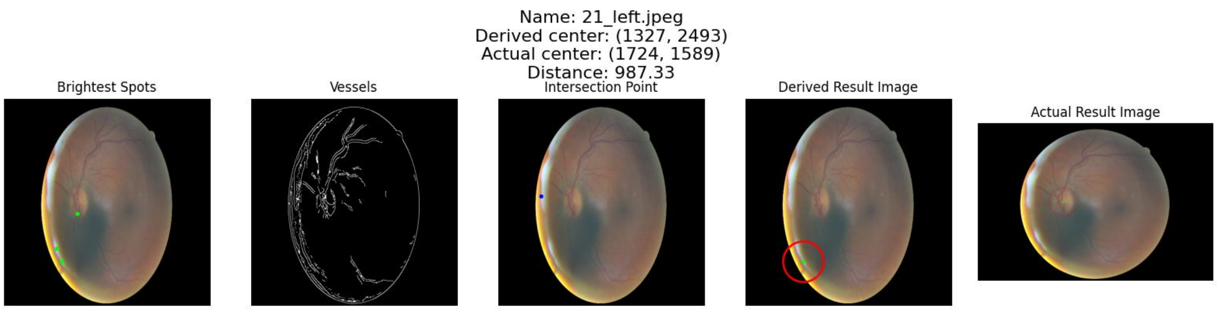
Name: 19_test.tif
Derived center: (483, 281)
Actual center: (486, 275)
Distance: 6.71

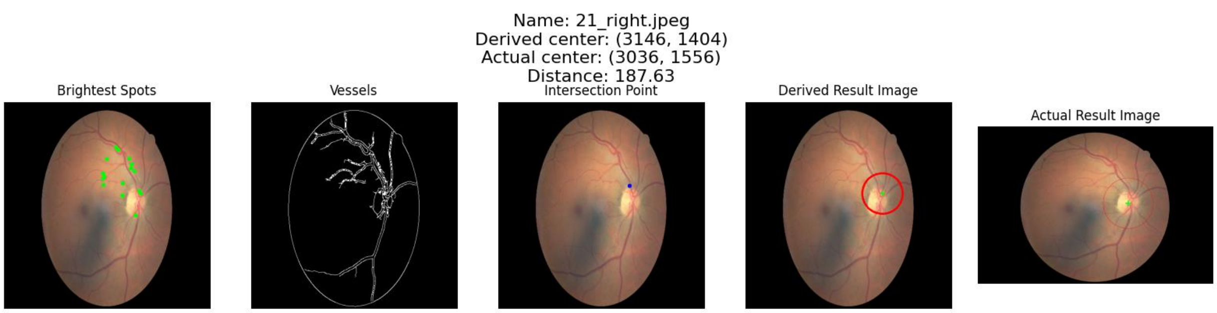
Name: 20_left.jpeg
Derived center: (2513, 1508)
Actual center: (2539, 1513)
Distance: 26.48

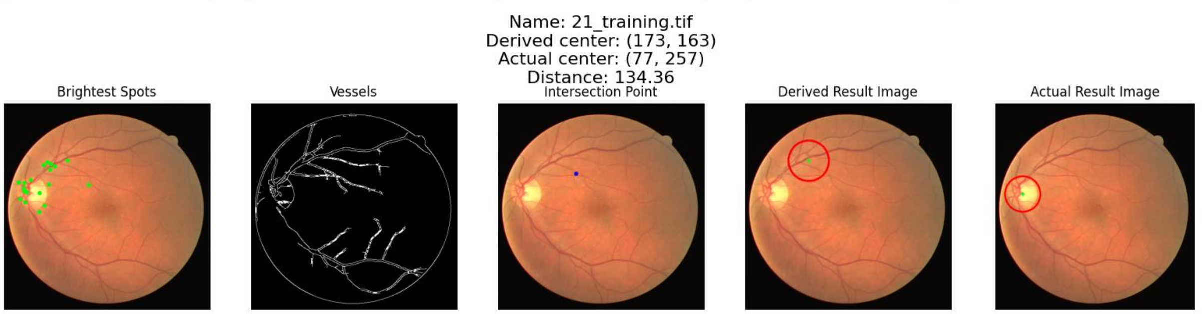



Name: 20_right.jpeg
Derived center: (1306, 1285)
Actual center: (1293, 1296)
Distance: 17.03
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 20_test.tif
Derived center: (482, 280)
Actual center: (482, 284)
Distance: 4.0
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 21_left.jpeg
Derived center: (1327, 2493)
Actual center: (1724, 1589)
Distance: 987.33
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

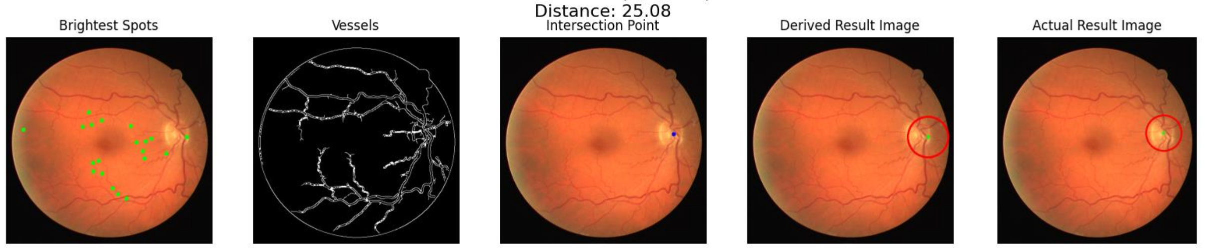

Name: 21_right.jpeg
Derived center: (3146, 1404)
Actual center: (3036, 1556)
Distance: 187.63
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 21_training.tif
Derived center: (173, 163)
Actual center: (77, 257)
Distance: 134.36
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

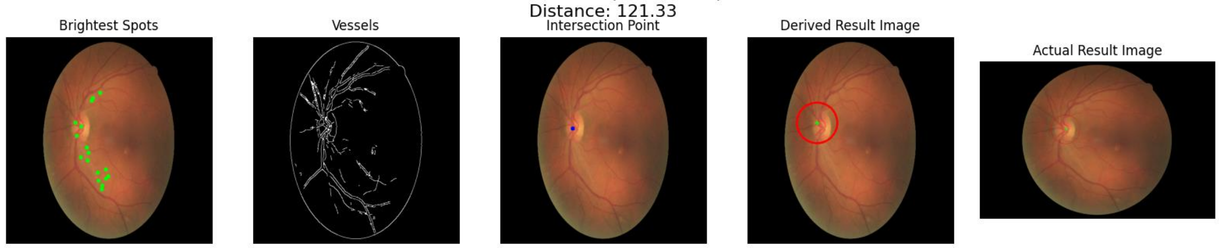
Name: 22_left.jpeg
Derived center: (810, 821)
Actual center: (747, 848)
Distance: 68.54

Name: 22_right.jpeg
Derived center: (1825, 907)
Actual center: (1821, 910)
Distance: 5.0

Name: 22_training.tif
Derived center: (494, 282)
Actual center: (471, 272)
Distance: 25.08

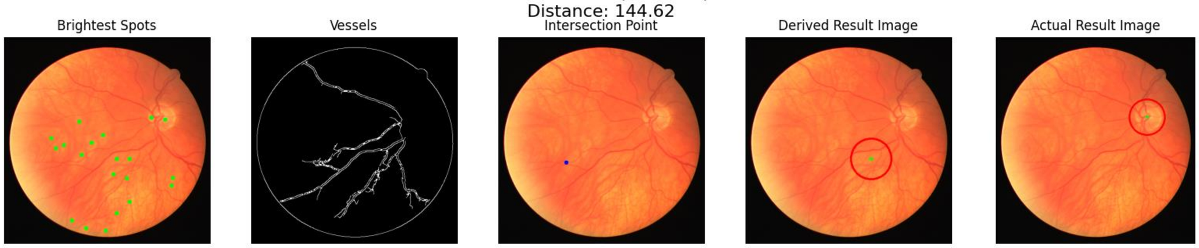


Name: 23_left.jpeg
Derived center: (1306, 1078)
Actual center: (1417, 1127)
Distance: 121.33



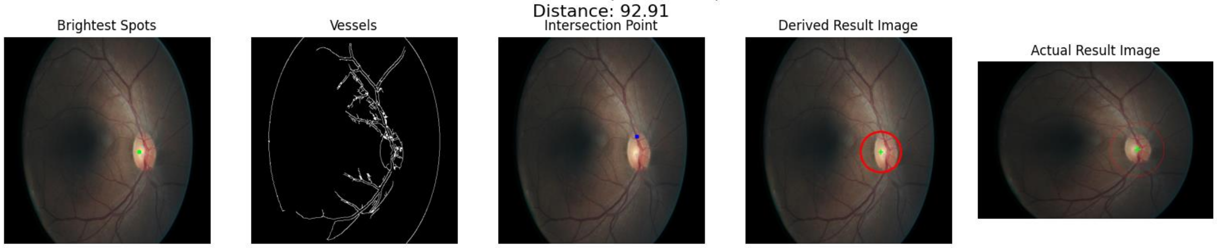
Name: 23_right.jpeg
Derived center: (2042, 577)
Actual center: (2676, 1242)
Distance: 918.79

Name: 23_training.tif
Derived center: (343, 344)
Actual center: (428, 227)
Distance: 144.62

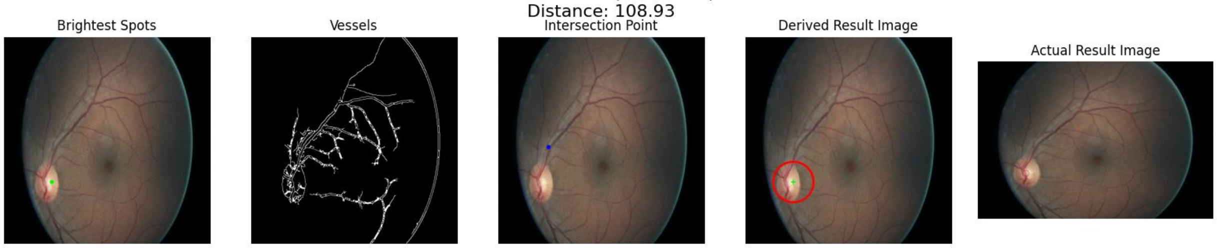


Name: 24_training.tif
Derived center: (484, 257)
Actual center: (472, 289)
Distance: 34.18

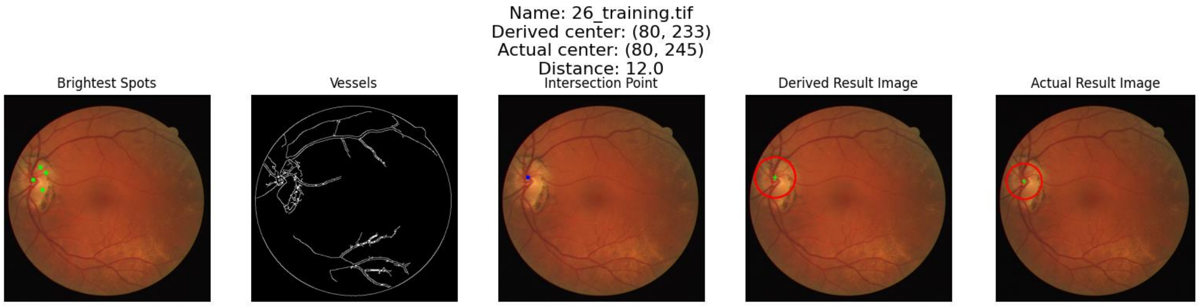
Name: 25_left.jpeg
Derived center: (2543, 1442)
Actual center: (2635, 1455)
Distance: 92.91

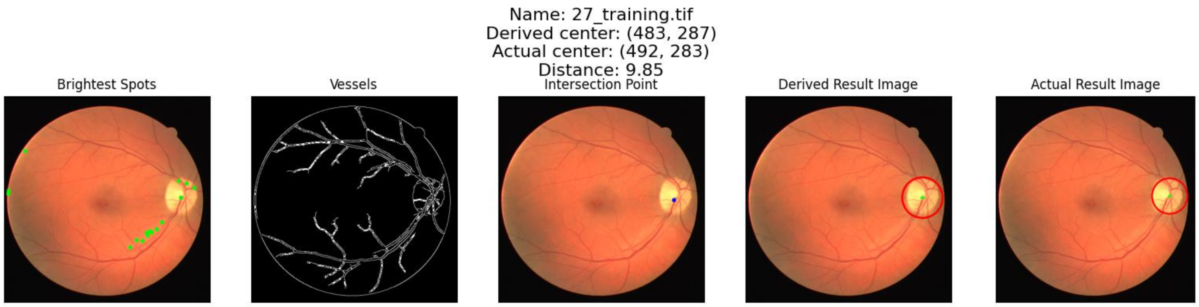


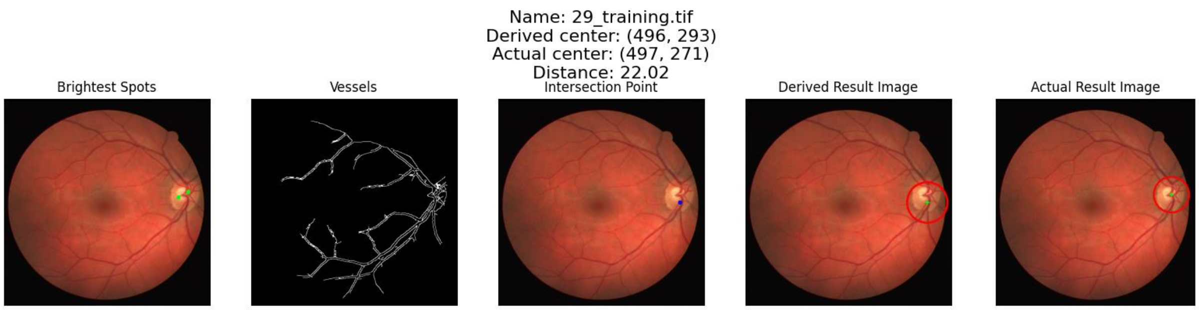
Name: 25_right.jpeg
Derived center: (896, 1817)
Actual center: (791, 1846)
Distance: 108.93

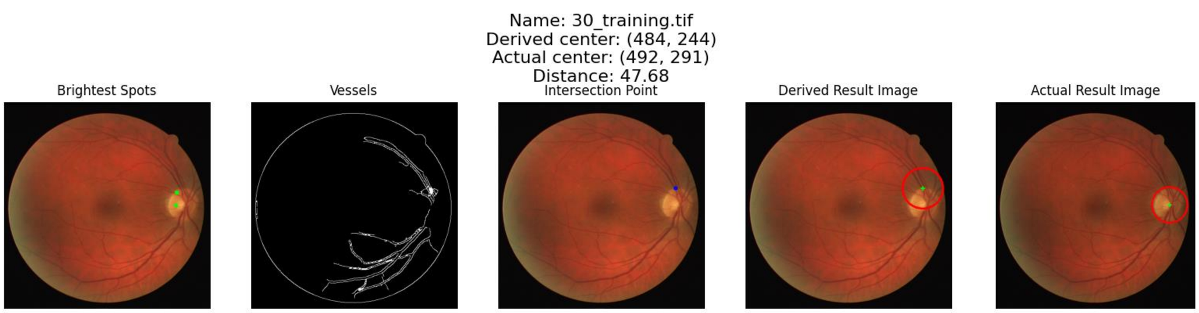


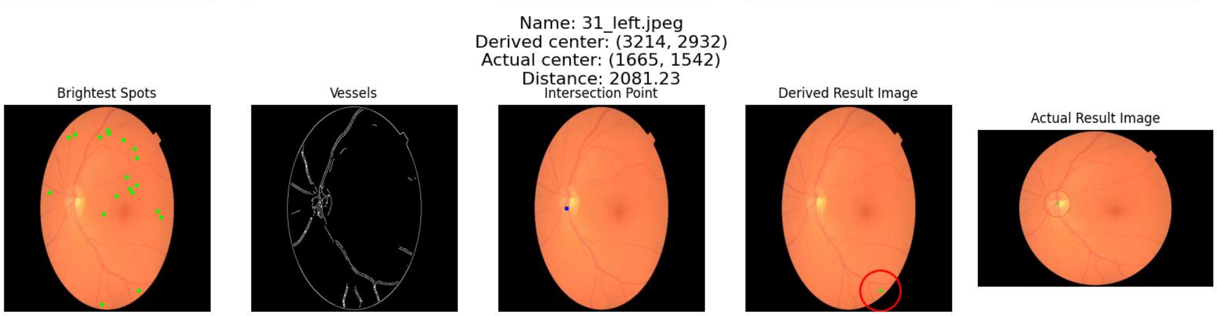
Name: 25_training.tif
Derived center: (466, 278)
Actual center: (464, 270)
Distance: 8.25

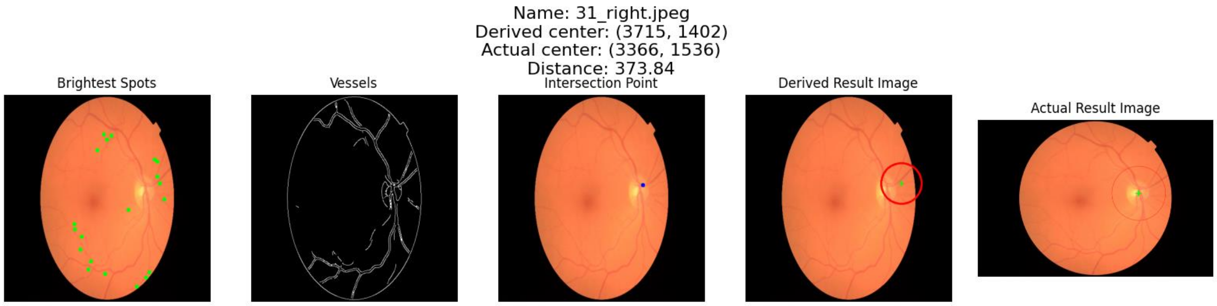

Name: 26_training.tif
Derived center: (80, 233)
Actual center: (80, 245)
Distance: 12.0
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

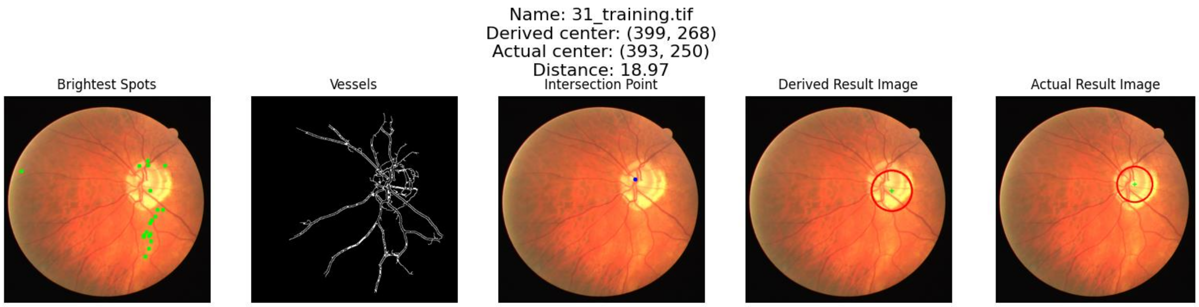

Name: 27_training.tif
Derived center: (483, 287)
Actual center: (492, 283)
Distance: 9.85
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

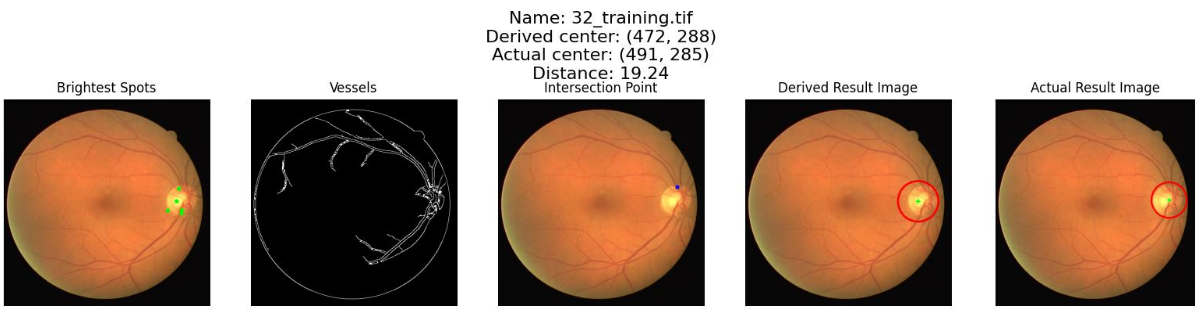

Name: 29_training.tif
Derived center: (496, 293)
Actual center: (497, 271)
Distance: 22.02
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

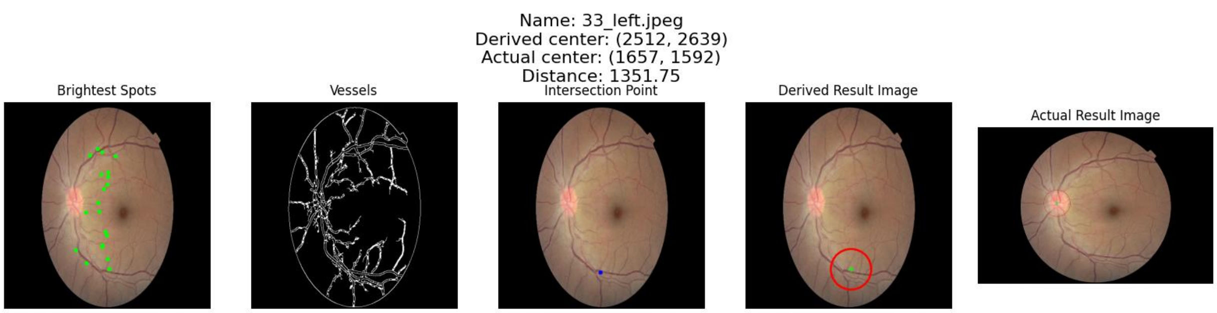

Name: 30_training.tif
Derived center: (484, 244)
Actual center: (492, 291)
Distance: 47.68
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

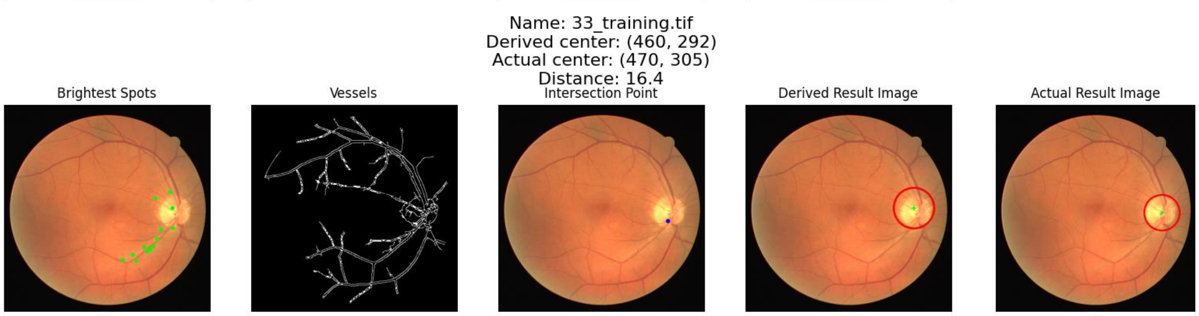

Name: 31_left.jpeg
Derived center: (3214, 2932)
Actual center: (1665, 1542)
Distance: 2081.23
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image

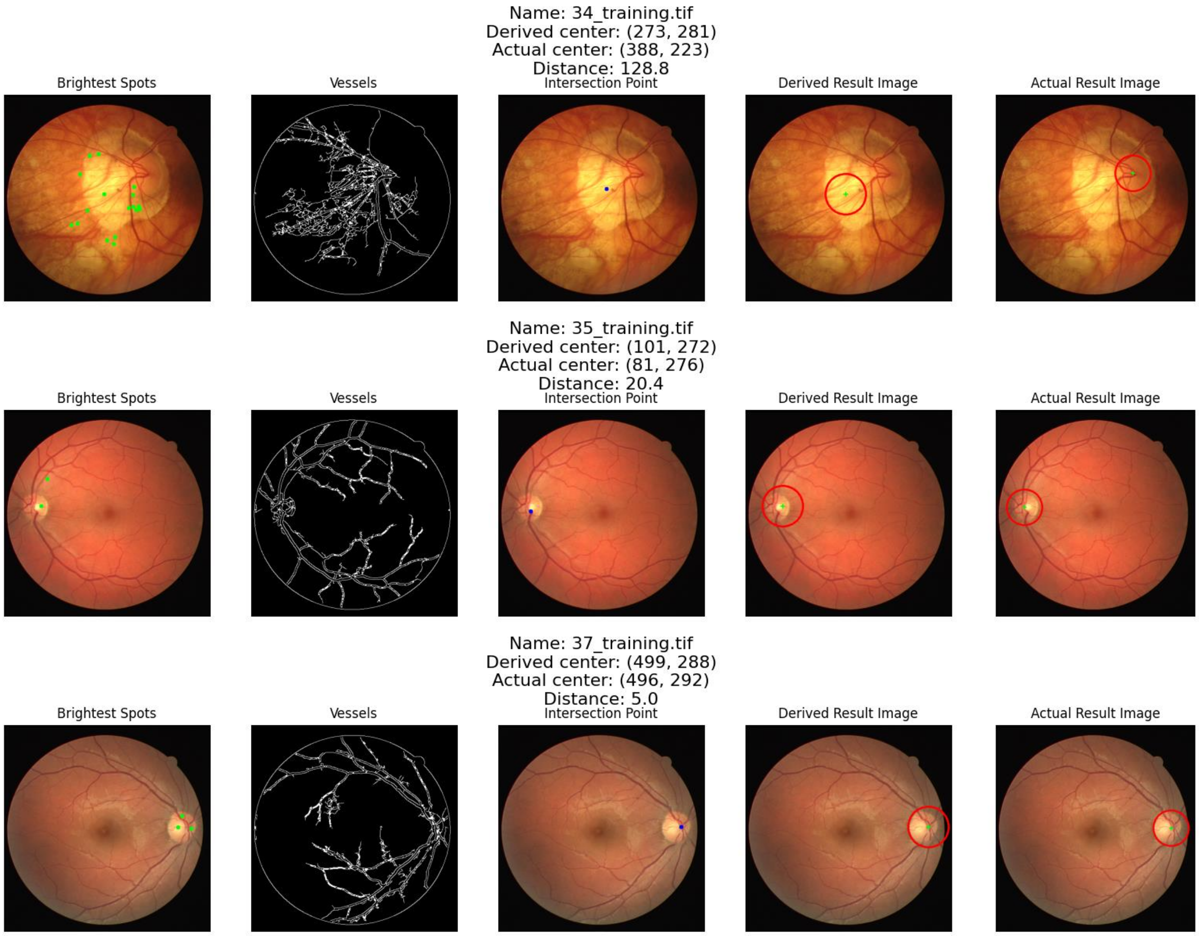

Name: 31_right.jpeg
Derived center: (3715, 1402)
Actual center: (3366, 1536)
Distance: 373.84
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 31_training.tif
Derived center: (399, 268)
Actual center: (393, 250)
Distance: 18.97
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 32_training.tif
Derived center: (472, 288)
Actual center: (491, 285)
Distance: 19.24
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 33_left.jpeg
Derived center: (2512, 2639)
Actual center: (1657, 1592)
Distance: 1351.75
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image


Name: 33_training.tif
Derived center: (460, 292)
Actual center: (470, 305)
Distance: 16.4
Brightest Spots
Vessels
Intersection Point
Derived Result Image
Actual Result Image



Code:

```
import os
import cv2
import csv
import numpy as np
from matplotlib.image import imread
import matplotlib.pyplot as plt

def read_images_from_folders(folder_path):
    images = []
    names = []
    files = os.listdir(folder_path)
    files.sort()
    for file in files:
        path = os.path.join(folder_path, file)
        names.append(file)
        path = path.replace("\\", "/")
        images.append(path)
```

```
    return images,names

def read_image_centers(file_path):
    centers = []
    with open(file_path, 'r') as file:
        for line in file:
            line = line.strip()
            if line:
                parts = line.split(',')
                if len(parts) == 3:
                    x = int(parts[1].strip())
                    y = int(parts[2].strip())
                    centers.append((x, y))
    return  centers

def pre_process_image(image, new_size=(512, 512)):
    original_size = image.shape[:2]
    target_size = new_size
    scaling_factor = (target_size[1] / original_size[1], target_size[0] / original_size[0])
    resized_image = cv2.resize(image, new_size)
    return resized_image, scaling_factor

def extract_vessels(image, min_component_area=2500):
    structure_element = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
    grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blur_image = cv2.GaussianBlur(grayscale_image, (5, 5), 0)
    edges = cv2.Canny(blur_image, 30, 20)
    connected_edges = cv2.morphologyEx(edges, cv2.MORPH_CLOSE, structure_element)
    num_labels, labels, stats, centroids = cv2.connectedComponentsWithStats(connected_edges, connectivity=8,
ltype=cv2.CV_32S)
    for label in range(1, num_labels):
        if stats[label, cv2.CC_STAT_AREA] < min_component_area:
            connected_edges[labels == label] = 0
    output = connected_edges
    num_vessels = cv2.countNonZero(connected_edges)
    # print('number of vessels found: ', num_vessels)
    if num_vessels < 6000:
        if min_component_area < 500:
            # print('less vessels found, s < 500 ')
            return output,num_vessels
        elif min_component_area < 1100:
            # print('less vessels found, s < 1100')
            output,num_vessels = extract_vessels(image, min_component_area-500)
        else:
            # print('less vessels found, s = 2500')
```

```
        output,num_vessels = extract_vessels(image, min_component_area-1500)

    return output, num_vessels

def find_brightest_spots(image):
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred_image = cv2.GaussianBlur(gray_image, (5, 5), 0)
    max_intensity = np.amax(blurred_image)
    _, thresholded_image = cv2.threshold(blurred_image, max_intensity * 0.7, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresholded_image, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    contours = sorted(contours, key=cv2.contourArea, reverse=True)[:20]

    bright_spots = []
    for contour in contours:
        x, y, w, h = cv2.boundingRect(contour)
        center_x = x + w // 2
        center_y = y + h // 2
        bright_spots.append((center_x, center_y))

    return bright_spots

def find_intersection_point(image):
    structure_element = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (7, 7))
    closed_image = cv2.morphologyEx(image, cv2.MORPH_CLOSE, structure_element)
    eroded_image = cv2.erode(closed_image, structure_element)
    contours, _ = cv2.findContours(eroded_image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    if contours:
        largest_contour = max(contours, key=cv2.contourArea)
        M = cv2.moments(largest_contour)
        if M["m00"] != 0:
            center_x = int(M["m10"] / M["m00"])
            center_y = int(M["m01"] / M["m00"])
            intersection_point = (center_x, center_y)
            return intersection_point
    return None

def analyze_data(image, brightest_spots,intersection_point,count, radius):
    max_vessel_count = -1
    list_of_info = []
    skip = False
    for spot in brightest_spots:
        info = {
            "center": (0, 0),
            "intensity": 0,
```

```
        "vessel_count": 0,
        "distance_from_intersection": 0,
        "neighbor_spots": 0,
    }

    masked_image = mask_circle(image, spot, radius)
    center = spot
    intensity = np.mean(masked_image)
    vessel_count = cv2.countNonZero(masked_image)
    # print("Vessel Count: ",vessel_count, "Count: ", count)
    distance_from_intersection = float('inf')
    neighbor_spots = 0;

    if intersection_point is not None:
        distance_from_intersection = np.sqrt((spot[0] - intersection_point[0])**2 + (spot[1] -
intersection_point[1])**2)
    if distance_from_intersection < radius:

        for bright_spot in brightest_spots:
            if bright_spot != spot:
                distance = float('inf')
                if intersection_point is not None:
                    distance = np.sqrt((bright_spot[0] - intersection_point[0])**2 + (bright_spot[1] -
intersection_point[1])**2)
                if distance < radius:
                    neighbor_spots += 1

        if neighbor_spots > len(brightest_spots)/2.5:
            center = intersection_point
            skip = True

    if skip is not True:
        if vessel_count > max_vessel_count:
            max_vessel_count = vessel_count

    info["center"] = center
    info["intensity"] = intensity
    info["vessel_count"] = vessel_count
    info["distance_from_intersection"] = distance_from_intersection
    info["neighbor_spots"] = neighbor_spots
    list_of_info.append(info)
  return  list_of_info

def find_best_spot(info):
```

```
    info = sorted(info, key=lambda x: (x["vessel_count"], x['intensity'], x["distance_from_intersection"],
x['neighbor_spots']), reverse=True)
    return info[0]["center"],info

def circle_brightest_spot(image, brightest_spot, radius):
    image = image.copy()
    cv2.circle(image, brightest_spot, radius, (0, 0, 255), 4)
    cross_size = int(radius/10)
    cross_thickness = int(radius/25)
    cross_color = (0, 255, 0)
    center_x, center_y = brightest_spot
    cv2.line(image, (center_x - cross_size, center_y), (center_x + cross_size, center_y), cross_color,
cross_thickness)
    cv2.line(image, (center_x, center_y - cross_size), (center_x, center_y + cross_size), cross_color,
cross_thickness)


    return image

def mask_circle(image, brightest_spot, radius):
    mask = np.zeros_like(image)
    cv2.circle(mask, brightest_spot, radius, (255, 255, 255), -1)
    masked_image = cv2.bitwise_and(image, mask)
    return masked_image

def draw_brightest_spots(image, brightest_spots):
    for spot in brightest_spots:
        cv2.circle(image, spot, 5, (0, 255, 0), -1)

def display(images, title,name, save_path):
    plt.figure(figsize=(16, 4))
    for i in range(len(images)):
        plt.subplot(1, len(images), i+1)
        plt.imshow(cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB), aspect='equal')
        if i == 0:
            plt.title('Brightest Spots')
        elif i == 1:
            plt.title('Vessels')
        elif i == 2:
            plt.title('Intersection Point')
        elif i == 3:
            plt.title('Derived Result Image')
        else:
            plt.title('Actual Result Image')
        plt.suptitle(title, fontsize=16)
```

```
        plt.axis('off')
    plt.tight_layout()
    plt.savefig(save_path + "/" + name + ".png" )
    plt.show()

def print_info_table(list_of_info):
    print("Optic Disk Information:")
    print("{:<15} {:<15} {:<15} {:<15} {:<15} ".format("Center", "Intensity", "Vessel Count", "Dst from IP", 
"Nbr Spots"))
    for info in list_of_info:
        center = info["center"]
        intensity =  round(info["intensity"], 2)
        vessel_count = info["vessel_count"]
        distance_from_intersection = round(info["distance_from_intersection"])
        neighbor_spots = info["neighbor_spots"]

        center_str = f"({center[0]}, {center[1]})"
        print("{:<15} {:<15} {:<15} {:<15} {:<15} ".format(center_str,intensity, vessel_count, 
distance_from_intersection, neighbor_spots,))
    print('----------------------------------------------------')


images,names = read_images_from_folders("Assignment-2/Fundus-image")
actual_centers = read_image_centers("Assignment-2/optic_disc_centres.csv")
derived_centers = []
distances = []
radius = 50
# images = [image for image,name in zip(images,names) if "test" in name]
# names = [name for name in names if "test" in name]

# images = [image for image,name in zip(images,names) if "1ffa" in name]
# names = [name for name in names if "1ffa" in name]

# images = images[:1]
# names = names[:1]

for path,name,center in zip(images,names,actual_centers):
    image = cv2.imread(path)
    preprocessed_image,scaling_factor = pre_process_image(image)
    intersection_point_image = preprocessed_image.copy()
    image_with_spots = preprocessed_image.copy()
    vessels,num_vessels = extract_vessels(preprocessed_image)
    brightest_spots = find_brightest_spots(preprocessed_image)
    intersection_point = find_intersection_point(vessels)
    if intersection_point:
```

```
        cv2.circle(intersection_point_image, intersection_point, 5, (255, 0, 0), -1)
    info = analyze_data(vessels, brightest_spots,intersection_point,num_vessels, radius)
    best_spot,info = find_best_spot(info)
    derived_result_image = circle_brightest_spot(preprocessed_image, best_spot, radius)
    draw_brightest_spots(image_with_spots, brightest_spots)
    best_spot_scaled = (int( best_spot[0]/scaling_factor[0] ), int( best_spot[1]/scaling_factor[1] ))
    actual_radius = radius
    if center[0] > 1000:
        actual_radius = int(radius * (center[0]/300))
    actual_result_image = circle_brightest_spot(image, center, actual_radius)
    distance = round(np.sqrt((best_spot_scaled[0] - center[0])**2 + (best_spot_scaled[1] - center[1])**2),2)
    distances.append(distance)
    derived_centers.append(best_spot_scaled)
    display([image_with_spots, vessels, intersection_point_image,
derived_result_image,actual_result_image],'Name: ' +  name + '\nDerived center: ' + str(best_spot_scaled) +
'\nActual center: ' + str(center) + '\nDistance: ' + str(distance),name,"Assignment-2/Result" )
    print_info_table(info)

print("Distances:")
print("{:<20} {:<20} {:<20} {:<20}".format("Image", "Derived OC", "Actual OC", "Distance"))
for (name, derived_center, actual_center, distance) in zip(names,derived_centers, actual_centers,distances):
    print("{:<20} {:<20} {:<20} {:<20}".format(name, str(derived_center), str(actual_center) , str(distance)))
with open("Assignment-2/resultant_optic_disc_centres.csv", mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["Image", "Derived OC", "Actual OC", "Distance"])
    for name, derived_center, actual_center, distance in zip(names, derived_centers, actual_centers, distances):
        writer.writerow([name, str(derived_center), str(actual_center), str(distance)])
print("Results have been written to", "Assignment-2/resultant_optic_disc_centres.csv")
```