

**Maaz**

List of preprocessing algorithms:

1. Gabor-based filters to enhance image quality
2. Wavelet transform to extract features from RFI
3. Noise elimination
4. Gamma correction for green channel extraction
5. Enhancement using CLAHE
6. Normalization
7. Noise removal technique
8. Adaptive filtering based on green histogram channel equalization for analyzing blood vessels in images
9. Gaussian filtering
10. Enhancing contrast
11. Applying thresholding
12. Removing backgrounds
13. Tophat Transformation (White Tophat)
14. BottomHAT Transformations (Black Tophat)
15. Extracting the green channel
16. Detecting contours
17. Performing morphological operations
18. Image resizing
19. Multiscale Retinex with Chromaticity Preservation (MSRCP)
20. The artificial bee colony (ABC) technique
21. Shape cropping
22. Cropping
23. Mix of mathematical morphology
24. Artifact removal
25. new radial invariant moments of Hahn in quaternion (QRHMI)
26. Rotation
27. Intensity normalization
28. Data normalization
29. ROI and median filtering extraction to eliminate noise
30. Optimal bandpass filter method
31. grayscale conversion
32. mask application to remove unnecessary dark areas
33. median filters
34. entropy thresholding
35. feature scaling

Non-repetitive list:

1. Gabor-based filters – enhance texture and edge details.
2. Wavelet transform – extract multi-scale features from RFI.
3. Noise elimination / Noise removal (combined term).
4. Gamma correction – applied especially for green channel enhancement.

5. Contrast Limited Adaptive Histogram Equalization (CLAHE) – improve local contrast.
6. Normalization / Data normalization / Intensity normalization (combined term).
7. Adaptive filtering based on green histogram equalization – analyze and enhance blood vessels.
8. Gaussian filtering – smoothen and reduce noise.
9. Thresholding (including entropy thresholding) – segment key regions.
10. Background removal / Mask application to remove unnecessary dark areas (combined term).
11. Top-hat Transformation (White Top-hat) – highlight bright structures.
12. Bottom-hat Transformation (Black Top-hat) – enhance dark structures.
13. Green channel extraction – isolate the most informative color component.
14. Contour detection – identify object or vessel boundaries.
15. Morphological operations – refine shapes, remove small artifacts.
16. Image resizing – standardize input dimensions.
17. Multiscale Retinex with Chromaticity Preservation (MSRCP) – improve lighting and color consistency.
18. Artificial Bee Colony (ABC) technique – optimization or feature selection.
19. Shape cropping / Cropping / ROI extraction (combined term).
20. Mathematical morphology – structure-based enhancement or feature extraction.
21. Artifact removal – eliminate image imperfections.
22. Radial invariant moments of Hahn in quaternion (QRHMI) – extract rotation-invariant shape features.
23. Rotation – alignment or data augmentation.
24. Median filtering / ROI and median filtering extraction (combined term).
25. Optimal bandpass filter – isolate specific frequency components.
26. Feature scaling – standardize numerical feature values.

Previously implemented combinations:

| Dataset                    | Pre-processing Technique Applied   |
|----------------------------|--|
| OCTA vascular density maps | Gabor-based filters to enhance the image quality [29], wavelet transform to extract features from RFI [31]   |
| DRIVE                      | Noise elimination, followed by green channel extraction and enhancement using CLAHE and gamma correction [61], normalization and noise removal techniques are applied and later an adaptive filtering based on green histogram channel equalization is employed for analyzing blood vessels in images [53], gaussian filtering and adaptive histogram equalization [35], enhancing contrast, applying thresholding, removing backgrounds, extracting the green channel, detecting contours, and performing morphological operations [62] |
| EyePACS                    | Image resizing, green channel extraction, Tophat Transformation (White Tophat), and BottomHAT Transformations (Black Tophat) were used [37], noise removal and contrast enhancement of the input images [32], The artificial bee colony (ABC) technique [63], shape cropping, cropping, and Gaussian blurring [64]   |
| APTOPS                     | Noise removal, artifact removal, CLAHE, and image resizing [65], new radial invariant moments of Hahn in quaternion (QRHMI) was employed [66], Gaussian filter for coherence, and image transformations like resizing, rotation, and color adjustments [34], green channel, CLAHE [33], CLAHE [67], [68]   |
| STARE                      | Noise reduction and local contrast enhancement [69], CLAHE [54]  |
| IDRiD                      | Intensity normalization and HE [70], data normalization, histogram equalization, and proportional resizing [71], RoI and median filtering extraction to eliminate noise [72], optimal bandpass filter method [73]  |
| MESSIDOR                   | CLAHE and Gaussian filtering [74], image resizing, noisy image removal [75], blurring, grayscale conversion, and mask application to remove unnecessary dark areas [76], CLAHE [77], filtering and green channel extraction techniques [27] Gabor filtering (GF) for noise reduction [78], CLAHE [79], CLAHE and HE [80] Multiscale Retinex with Chromaticity Preservation (MSRCP) for improving image contrast [81], noise channel and median filters, CLAHE [82]   |
| DIARETDB1                  | Mix of mathematical morphology, Gaussian match filtering, and entropy thresholding [83], normalization [84], feature scaling, masking, and resizing [36]   |
| CHASE                      | Normalization [85]   |

## 1. Blood Vessel Analysis / Retinal Image Enhancement

Focus: green-channel-based enhancement, adaptive equalization, and vessel extraction.

### Pipeline A – *Green-based Vessel Enhancement*

1. **Green channel extraction**
2. **Gamma correction** (tune brightness on green component)
3. **Adaptive filtering based on green histogram equalization**
4. **Gaussian filtering** (reduce fine noise)
5. **CLAHE** (enhance local contrast)
6. **Top-hat transformation** (highlight vessels)

7. **Bottom-hat transformation** (remove background shadows)
8. **Morphological operations** (cleanup, refine structures)
9. **Thresholding** (segment vessels)
10. **Contour detection** (extract vessel edges)

→ Use-case: Retinal fundus image preprocessing for vessel segmentation.

---

### **Pipeline B – Adaptive & Optimized Vessel Detection**

1. **MSRCP** (fix illumination/color balance)
2. **Green channel extraction**
3. **Adaptive filtering based on green histogram equalization**
4. **Gabor-based filters** (enhance directional vessels)
5. **Median filtering** (remove impulsive noise)
6. **Thresholding (entropy-based)**
7. **Morphological operations + Mathematical morphology**
8. **Contour detection + Shape cropping (ROI)**
9. **Feature scaling** (for ML models)

→ Use-case: Automated retinal vessel detection or diabetic retinopathy preprocessing.

---

## **2. RFI (Radio Frequency Interference) or Multi-scale Feature Extraction**

Focus: texture, frequency, and multi-resolution analysis.

### **Pipeline C – Wavelet + Feature Scaling**

1. **Background removal** (normalize intensity map)
2. **Wavelet transform** (extract multi-scale RFI features)
3. **Optimal bandpass filter** (isolate signal frequency bands)
4. **Noise elimination**
5. **Normalization** (scale features)
6. **Thresholding (entropy or Otsu)** (extract interference zones)
7. **Contour detection + Shape cropping (ROI)**
8. **QRHMI** (radial invariant moment computation)
9. **Feature scaling**

→ Use-case: Signal-image-based RFI detection with rotation-invariant texture features.

---

#### **Pipeline D – Optimization-Aided RFI Feature Extraction**

1. **Wavelet transform**
2. **Optimal bandpass filter**
3. **Artificial Bee Colony (ABC) optimization** (auto-tune thresholds/filter parameters)
4. **Gabor-based filtering** (enhance texture features)
5. **Normalization**
6. **QRHMI** (rotation-invariant feature extraction)
7. **Feature scaling**

→ Use-case: Automated RFI or pattern classification with optimization-based feature tuning.

---



### 3. General Image Enhancement / Denoising

Focus: removing artifacts, enhancing contrast, preparing clean input.

#### Pipeline E – *General Enhancement + Normalization*

1. **Background removal / Mask application**
2. **Noise elimination / Median filtering**
3. **Gaussian filtering**
4. **CLAHE** (local contrast improvement)
5. **Gamma correction**
6. **Normalization**
7. **Morphological operations**
8. **Artifact removal**
9. **Image resizing**

→ Use-case: Preprocessing medical or photographic images before segmentation/classification.

---

#### Pipeline F – *Contrast and Texture Refinement*

1. **MSRCP** (lighting normalization)
2. **Gabor-based filters** (enhance texture and edges)
3. **CLAHE**
4. **Top-hat transformation**
5. **Bottom-hat transformation**
6. **Normalization**

7. **Thresholding**
8. **Artifact removal**
9. **Shape cropping (ROI extraction)**

→ Use-case: General-purpose contrast enhancement and edge feature emphasis.

---

## 4. Feature Extraction & Invariant Representation

Focus: obtaining descriptors robust to scale, rotation, and illumination.

### **Pipeline G – *Invariant Feature Descriptor Creation***

1. **Normalization / Intensity normalization**
2. **Gabor-based filtering**
3. **Wavelet transform**
4. **QRHMI computation** (extract rotation-invariant features)
5. **Feature scaling**

→ Use-case: Classification or recognition tasks needing rotation/scale invariance.

---

### **Pipeline H – *Optimized Morphological Feature Extraction***

1. **Mathematical morphology** (structural enhancement)
2. **Top-hat + Bottom-hat** transformations
3. **Artificial Bee Colony (ABC)** optimization (for feature or threshold selection)
4. **Thresholding + Contour detection**

## 5. Feature scaling

→ Use-case: Adaptive structure-based pattern recognition or defect detection.

---

# ⌚ 5. Preprocessing & Augmentation Pipelines

Focus: preparing datasets for ML / Deep Learning.

### Pipeline I – *Clean Input Preparation*

1. Background removal / Mask application
2. Noise elimination (Median/Gaussian)
3. Artifact removal
4. Cropping / Shape cropping / ROI extraction
5. Rotation (augmentation)
6. Image resizing
7. Normalization
8. Feature scaling

→ Use-case: Building standardized, noise-free input sets for CNNs or SVMs.

---

# ⚙️ 6. Compact Reference (Building Block Sets)

| Purpose                    | Minimal Valid Combo  |
|----------------------------|--|
| Edge & texture enhancement | Gabor-based filters → CLAHE → Normalization                            |
| Vessel enhancement         | Green channel extraction → Adaptive filtering → Top-hat → Thresholding |

|                             |  |
|-----------------------------|--|
| RFI features                | Wavelet transform → Optimal bandpass filter → QRHMI<br>→ Feature scaling |
| Artifact removal            | Mask application → Morphological operations →<br>Artifact removal        |
| Illumination correction     | MSRCP → Gamma correction → CLAHE   |
| Noise reduction             | Median filtering → Gaussian filtering → Normalization                    |
| ROI preparation             | Cropping → Resize → Normalize  |
| Optimized feature selection | ABC technique → Feature scaling  |

---

## 7. Diabetic Retinopathy Detection - Optimized Pipelines

Focus: Comprehensive preprocessing for robust DR detection, targeting specific DR signs like microaneurysms, hemorrhages, and exudates.

### Pipeline J – Advanced DR Feature Enhancement

1. **MSRCP** (normalize lighting and color, crucial for varied fundus images)
2. **Green channel extraction** (maximizes contrast for vessel and lesion analysis)
3. **Adaptive filtering based on green histogram equalization** (enhances vessel and microaneurysm visibility)
4. **CLAHE** (improves local contrast of lesions)
5. **Gaussian filtering** (initial noise reduction, preserving edges)
6. **Top-hat Transformation** (highlights bright lesions like exudates and cotton wool spots)
7. **Bottom-hat Transformation** (enhances dark lesions like hemorrhages and microaneurysms)
8. **Morphological operations** (refine lesion shapes, remove small artifacts)
9. **Thresholding (entropy-based)** (segments lesions based on information content)
10. **Contour detection + Shape cropping (ROI extraction)** (isolates detected lesions for further analysis)
11. **Feature scaling** (prepares extracted features for machine learning classification)

→ Use-case: Maximizing visibility and extracting features of various DR lesions for robust classification models.

### Pipeline K – Denoised & Normalized Input for Deep Learning DR

1. **Background removal / Mask application** (removes irrelevant dark areas, focusing on the retina)

2. **Noise elimination (Median/Gaussian filtering)** (reduces imaging noise while preserving details)
3. **Artifact removal** (eliminates dust, reflections, or other non-physiological elements)
4. **Cropping / Shape cropping / ROI extraction** (focuses on the relevant retinal area, e.g., macula-optic disc region)
5. **Image resizing** (standardizes input dimensions for deep learning models)
6. **Rotation** (data augmentation to improve model generalization)
7. **Normalization** (standardizes pixel intensity values across the dataset)
8. **Feature scaling** (ensures uniform data distribution for deep learning model stability)

→ Use-case: Preparing a clean, augmented, and standardized dataset for training convolutional neural networks (CNNs) for DR classification.

TODO\_Mz

Entry: 19-10-2025 16:11

From the research paper - Application of Artificial Intelligence for Classification Segmentation Early Detection Early Diagnosis and Grading of Diabetic Retinopathy From Fundus Retinal Images A Comprehensive Review, apply all preprocessing algorithms one-by-one on 100 images from ODIR-5K

Deadline: 19-10-2025

# Combos

# 1. Blood Vessel Analysis / Retinal Image Enhancement

Focus: green-channel-based enhancement, adaptive equalization, and vessel extraction.

## Pipeline A – *Green-based Vessel Enhancement*

1. **Green channel extraction**
2. **Gamma correction** (tune brightness on green component)
3. **Adaptive filtering based on green histogram equalization**
4. **Gaussian filtering** (reduce fine noise)
5. **CLAHE** (enhance local contrast)
6. **Top-hat transformation** (highlight vessels)
7. **Bottom-hat transformation** (remove background shadows)
8. **Morphological operations** (cleanup, refine structures)
9. **Thresholding** (segment vessels)
10. **Contour detection** (extract vessel edges)

→ Use-case: Retinal fundus image preprocessing for vessel segmentation.

---

## Pipeline B – *Adaptive & Optimized Vessel Detection*

1. **MSRCP** (fix illumination/color balance)
2. **Green channel extraction**
3. **Adaptive filtering based on green histogram equalization**
4. **Gabor-based filters** (enhance directional vessels)

5. **Median filtering** (remove impulsive noise)
6. **Thresholding (entropy-based)**
7. **Morphological operations + Mathematical morphology**
8. **Contour detection + Shape cropping (ROI)**
9. **Feature scaling** (for ML models)

→ Use-case: Automated retinal vessel detection or diabetic retinopathy preprocessing.

---

## **2. RFI (Radio Frequency Interference) or Multi-scale Feature Extraction**

Focus: texture, frequency, and multi-resolution analysis.

### **Pipeline C – Wavelet + Feature Scaling**

1. **Background removal** (normalize intensity map)
2. **Wavelet transform** (extract multi-scale RFI features)
3. **Optimal bandpass filter** (isolate signal frequency bands)
4. **Noise elimination**
5. **Normalization** (scale features)
6. **Thresholding (entropy or Otsu)** (extract interference zones)
7. **Contour detection + Shape cropping (ROI)**
8. **QRHMI** (radial invariant moment computation)
9. **Feature scaling**

→ Use-case: Signal-image-based RFI detection with rotation-invariant texture features.

---

## **Pipeline D – Optimization-Aided RFI Feature Extraction**

1. **Wavelet transform**
2. **Optimal bandpass filter**
3. **Artificial Bee Colony (ABC)** optimization (auto-tune thresholds/filter parameters)
4. **Gabor-based filtering** (enhance texture features)
5. **Normalization**
6. **QRHMI** (rotation-invariant feature extraction)
7. **Feature scaling**

➡ Use-case: Automated RFI or pattern classification with optimization-based feature tuning.

---

## **3. General Image Enhancement / Denoising**

Focus: removing artifacts, enhancing contrast, preparing clean input.

## **Pipeline E – General Enhancement + Normalization**

1. **Background removal / Mask application**
2. **Noise elimination / Median filtering**
3. **Gaussian filtering**
4. **CLAHE** (local contrast improvement)
5. **Gamma correction**
6. **Normalization**
7. **Morphological operations**

## **8. Artifact removal**

## **9. Image resizing**

→ Use-case: Preprocessing medical or photographic images before segmentation/classification.

# Green-based Vessel Enhancement

# 1. Blood Vessel Analysis / Retinal Image Enhancement

Focus: green-channel-based enhancement, adaptive equalization, and vessel extraction.

## Pipeline A – *Green-based Vessel Enhancement*

1. **Green channel extraction**
2. **Gamma correction** (tune brightness on green component)
3. **Adaptive filtering based on green histogram equalization**
4. **Gaussian filtering** (reduce fine noise)
5. **CLAHE** (enhance local contrast)
6. **Top-hat transformation** (highlight vessels)
7. **Bottom-hat transformation** (remove background shadows)
8. **Morphological operations** (cleanup, refine structures)
9. **Thresholding** (segment vessels)
10. **Contour detection** (extract vessel edges)

→ Use-case: Retinal fundus image preprocessing for vessel segmentation.

```

import cv2
import numpy as np
import os
import pandas as pd
import matplotlib.pyplot as plt
from skimage.morphology import skeletonize
%matplotlib inline

#
=====

=====
# Directories
#
=====

=====
img_dir = "./datasets/training_images_limited" # input folder
out_dir = "./datasets/green_based_vessel_enhancement_type_1"
os.makedirs(out_dir, exist_ok=True)

#
=====

=====
# Helper Functions
#
=====

=====
def gamma_correction(image, gamma=1.0):
    """Apply gamma correction."""
    invGamma = 1.0 / gamma
    table = np.array([(i / 255.0) ** invGamma * 255 for i in
np.arange(256)]).astype("uint8")
    return cv2.LUT(image, table)

def measure_vessel_properties(binary_mask):
    """Calculate vessel area and approximate length (via skeleton)."""
    # Area: number of white pixels
    area = np.sum(binary_mask > 0)

    # Skeletonize for vessel length estimation (thin 1-pixel wide)
    skeleton = skeletonize(binary_mask > 0)
    vessel_length = np.sum(skeleton)

    return area, vessel_length, skeleton.astype(np.uint8) * 255

def process_vessel_enhancement(img_bgr):
    """Complete image processing pipeline for green-based vessel
enhancement."""
    # 1. Green channel extraction
    green = img_bgr[:, :, 1]

```

```

# 2. Gamma correction
gamma_corrected = gamma_correction(green, gamma=1.2)

# 3. Histogram equalization
hist_eq = cv2.equalizeHist(gamma_corrected)

# 4. Gaussian filtering
blurred = cv2.GaussianBlur(hist_eq, (5, 5), 0)

# 5. CLAHE
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
clahe_img = clahe.apply(blurred)

# 6. Top-hat & 7. Bottom-hat transformations
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (15, 15))
top_hat = cv2.morphologyEx(clahe_img, cv2.MORPH_TOPHAT, kernel)
bottom_hat = cv2.morphologyEx(clahe_img, cv2.MORPH_BLACKHAT,
kernel)

# Combine to improve vessel contrast
enhanced = cv2.add(clahe_img, top_hat)
enhanced = cv2.subtract(enhanced, bottom_hat)

# 8. Morphological cleanup
morph_kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
morph_clean = cv2.morphologyEx(enhanced, cv2.MORPH_OPEN,
morph_kernel)

# 9. Thresholding (Otsu)
_, thresh = cv2.threshold(morph_clean, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

# 10. Contour detection
contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
contour_mask = np.zeros_like(thresh)
cv2.drawContours(contour_mask, contours, -1, 255, 1)

# 11. Vessel measurements (area, length, skeleton)
area, length, skeleton_img = measure_vessel_properties(thresh)

return {
    "green": green,
    "enhanced": enhanced,
    "thresholded": thresh,
    "contour_mask": contour_mask,
    "skeleton": skeleton_img,
    "area": area,
    "length": length
}

```

```

#
=====
=====

# Batch Process All Images
#
=====

=====

measurements = []

for fname in os.listdir(img_dir):
    if fname.lower().endswith('.png', '.jpg', '.jpeg', '.tif'):
        path = os.path.join(img_dir, fname)
        img = cv2.imread(path)
        if img is None:
            continue

        results = process_vessel_enhancement(img)

        base = os.path.splitext(fname)[0]
        cv2.imwrite(os.path.join(out_dir, f"{base}_enhanced.png"),
results["enhanced"])
        cv2.imwrite(os.path.join(out_dir, f"{base}_segmented.png"),
results["thresholded"])
        cv2.imwrite(os.path.join(out_dir, f"{base}_contours.png"),
results["contour_mask"])
        cv2.imwrite(os.path.join(out_dir, f"{base}_skeleton.png"),
results["skeleton"])

        measurements.append({
            "filename": fname,
            "vessel_area_pixels": results["area"],
            "vessel_length_pixels": results["length"]
        })

    print(f"Processed: {fname}")

Processed: 0_left.jpg
Processed: 0_right.jpg
Processed: 10_left.jpg
Processed: 10_right.jpg
Processed: 11_left.jpg
Processed: 11_right.jpg
Processed: 12_left.jpg
Processed: 12_right.jpg
Processed: 13_left.jpg
Processed: 13_right.jpg
Processed: 14_left.jpg
Processed: 14_right.jpg
Processed: 15_left.jpg
Processed: 15_right.jpg
Processed: 16_left.jpg

```

Processed: 16\_right.jpg  
Processed: 17\_left.jpg  
Processed: 17\_right.jpg  
Processed: 18\_left.jpg  
Processed: 18\_right.jpg  
Processed: 19\_left.jpg  
Processed: 19\_right.jpg  
Processed: 1\_left.jpg  
Processed: 1\_right.jpg  
Processed: 20\_left.jpg  
Processed: 20\_right.jpg  
Processed: 21\_left.jpg  
Processed: 21\_right.jpg  
Processed: 22\_left.jpg  
Processed: 22\_right.jpg  
Processed: 23\_left.jpg  
Processed: 23\_right.jpg  
Processed: 24\_left.jpg  
Processed: 24\_right.jpg  
Processed: 25\_left.jpg  
Processed: 25\_right.jpg  
Processed: 26\_left.jpg  
Processed: 26\_right.jpg  
Processed: 27\_left.jpg  
Processed: 27\_right.jpg  
Processed: 28\_left.jpg  
Processed: 28\_right.jpg  
Processed: 29\_left.jpg  
Processed: 29\_right.jpg  
Processed: 2\_left.jpg  
Processed: 2\_right.jpg  
Processed: 30\_left.jpg  
Processed: 30\_right.jpg  
Processed: 31\_left.jpg  
Processed: 31\_right.jpg  
Processed: 32\_left.jpg  
Processed: 32\_right.jpg  
Processed: 33\_left.jpg  
Processed: 33\_right.jpg  
Processed: 34\_left.jpg  
Processed: 34\_right.jpg  
Processed: 35\_left.jpg  
Processed: 35\_right.jpg  
Processed: 36\_left.jpg  
Processed: 36\_right.jpg  
Processed: 37\_left.jpg  
Processed: 37\_right.jpg  
Processed: 38\_left.jpg  
Processed: 38\_right.jpg  
Processed: 39\_left.jpg  
Processed: 39\_right.jpg

```
Processed: 3_left.jpg
Processed: 3_right.jpg
Processed: 40_left.jpg
Processed: 40_right.jpg
Processed: 41_left.jpg
Processed: 41_right.jpg
Processed: 42_left.jpg
Processed: 42_right.jpg
Processed: 43_left.jpg
Processed: 43_right.jpg
Processed: 44_left.jpg
Processed: 44_right.jpg
Processed: 45_left.jpg
Processed: 45_right.jpg
Processed: 46_left.jpg
Processed: 46_right.jpg
Processed: 47_left.jpg
Processed: 47_right.jpg
Processed: 48_left.jpg
Processed: 48_right.jpg
Processed: 49_left.jpg
Processed: 49_right.jpg
Processed: 4_left.jpg
Processed: 4_right.jpg
Processed: 5_left.jpg
Processed: 5_right.jpg
Processed: 6_left.jpg
Processed: 6_right.jpg
Processed: 7_left.jpg
Processed: 7_right.jpg
Processed: 8_left.jpg
Processed: 8_right.jpg
Processed: 9_left.jpg
Processed: 9_right.jpg
```

```
#
=====
=====  
# Export Measurements
#
=====  
=====  
df = pd.DataFrame(measurements)
csv_path = os.path.join(out_dir, "vessel_measurements.csv")
df.to_csv(csv_path, index=False)
print(f"\n✓ Processing complete. Measurements saved to {csv_path}")
```

✓ Processing complete. Measurements saved to  
./datasets/green\_based\_vessel\_enhancement\_type\_1\vessel\_measurements.cs  
v

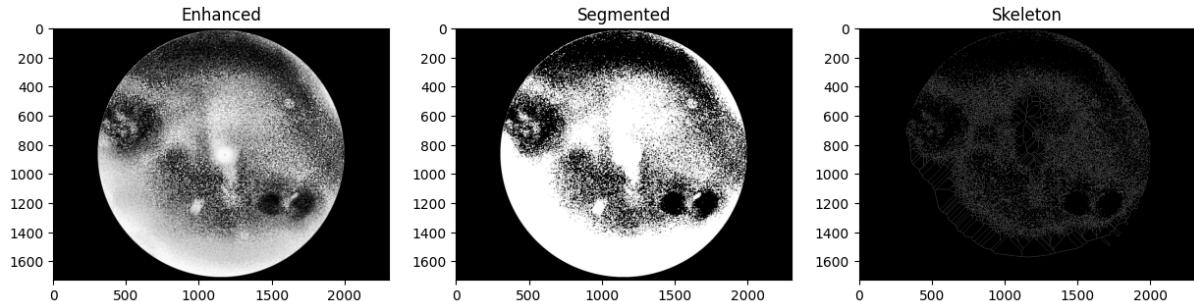
```

sample_file = os.listdir(img_dir)[0]
img_path = os.path.join(out_dir, os.path.splitext(sample_file)[0] +
"_enhanced.png")
seg_path = os.path.join(out_dir, os.path.splitext(sample_file)[0] +
"_segmented.png")
skeleton_path = os.path.join(out_dir, os.path.splitext(sample_file)[0] +
"_skeleton.png")

enh = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
seg = cv2.imread(seg_path, cv2.IMREAD_GRAYSCALE)
skeleton = cv2.imread(skeleton_path, cv2.IMREAD_GRAYSCALE)

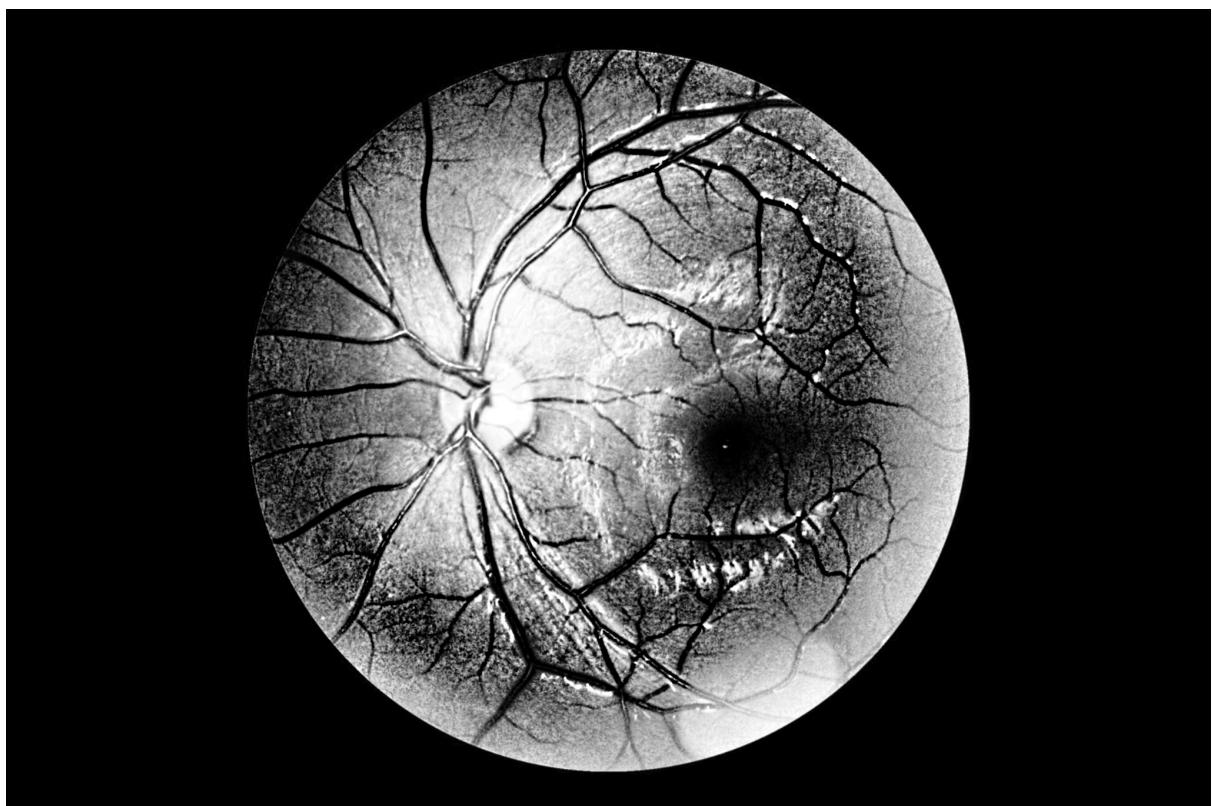
plt.figure(figsize=(15,5))
plt.subplot(1,3,1); plt.imshow(enh, cmap='gray'); plt.title('Enhanced')
plt.subplot(1,3,2); plt.imshow(seg, cmap='gray');
plt.title('Segmented')
plt.subplot(1,3,3); plt.imshow(skeleton, cmap='gray');
plt.title('Skeleton')
plt.show()

```





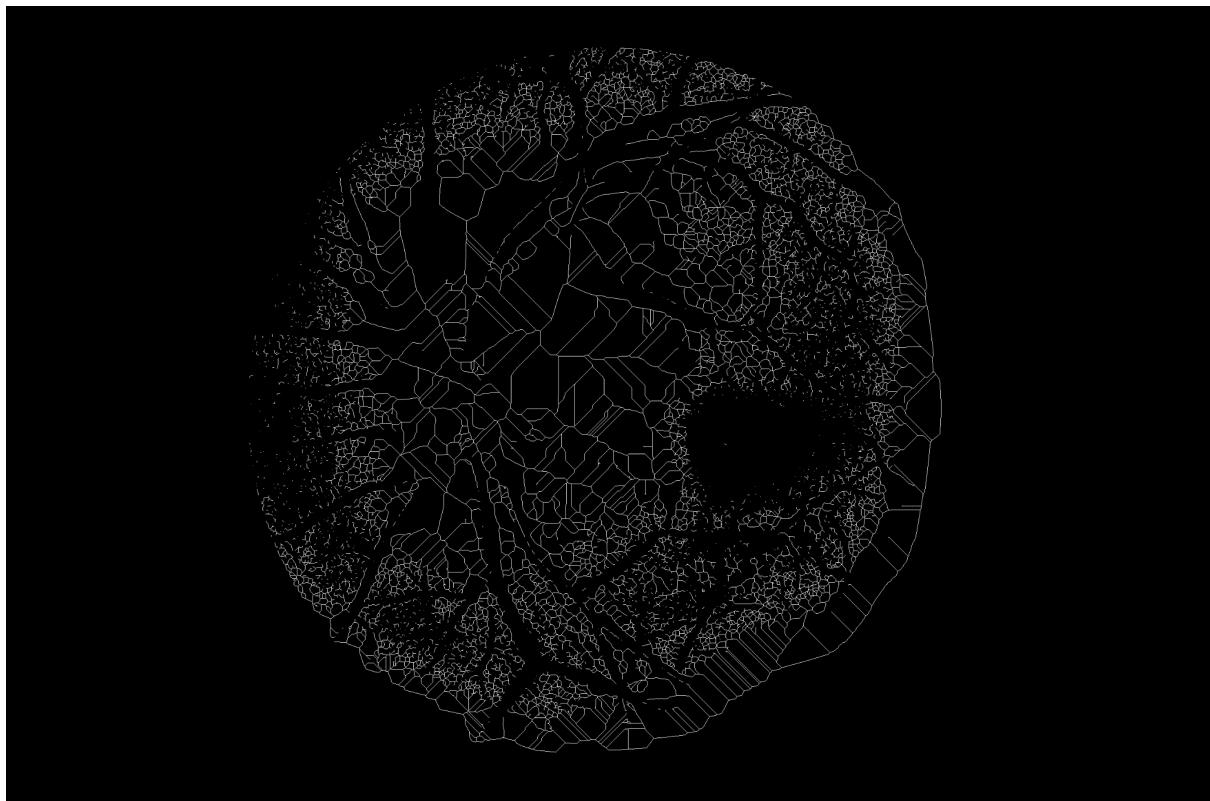
12\_left\_contours



12\_left\_enhanced



12\_left\_segmented



12\_left\_skeleton

# Adaptive & Optimized Vessel Detection

## **Pipeline B – Adaptive & Optimized Vessel Detection**

1. **MSRCP** (fix illumination/color balance)
2. **Green channel extraction**
3. **Adaptive filtering based on green histogram equalization**
4. **Gabor-based filters** (enhance directional vessels)
5. **Median filtering** (remove impulsive noise)
6. **Thresholding (entropy-based)**
7. **Morphological operations + Mathematical morphology**
8. **Contour detection + Shape cropping (ROI)**
9. **Feature scaling** (for ML models)

➡ Use-case: Automated retinal vessel detection or diabetic retinopathy preprocessing.

```
import cv2

import numpy as np

import os

import matplotlib.pyplot as plt

from skimage.filters import threshold_li, gabor

from skimage.morphology import remove_small_objects, skeletonize

from sklearn.preprocessing import MinMaxScaler

%matplotlib inline
```

```
# Directories

img_dir = "./datasets/training_images_limited"

out_dir = "./datasets/adaptive_optimized_vessel_detection"

os.makedirs(out_dir, exist_ok=True)
```

```
# Helper functions

def simplest_color_balance(img, percent=1):

    """Perform simple color balance similar to MSRCP step."""

    assert img.shape[2] == 3

    half_percent = percent / 200.0

    out_channels = []

    for channel in cv2.split(img):

        flat = channel.flatten()

        flat = np.sort(flat)

        n = len(flat)

        low_val = flat[int(n * half_percent)]

        high_val = flat[int(n * (1.0 - half_percent))]

        channel = np.clip(channel, low_val, high_val)

    return cv2.merge(out_channels)
```

```

        channel = (channel - low_val) / (high_val - low_val) * 255
        out_channels.append(channel)

    return cv2.merge(out_channels).astype(np.uint8)

def multi_scale_retinex(img, scales=[15, 80, 250], weight=None):
    """Simplified MSR implementation."""
    img = img.astype(np.float32) + 1.0

    if weight is None:
        weight = [1/len(scales)] * len(scales)

    retinex = np.zeros_like(img)

    for w, scale in zip(weight, scales):
        blur = cv2.GaussianBlur(img, (0,0), scale)

        retinex += w * (np.log10(img) - np.log10(blur))

    retinex = (retinex - np.min(retinex)) / (np.max(retinex) -
    np.min(retinex)) * 255.0

    return retinex.astype(np.uint8)

def MSRCP(img_bgr):
    """MSRCP (Multi-Scale Retinex with Color Preservation)."""
    img_bgr = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)
    img_retinex = multi_scale_retinex(img_bgr)
    return simplest_color_balance(img_retinex, percent=1)

def adaptive_histogram_equalization(img):
    """Adaptive histogram equalization on single-channel image."""
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    return clahe.apply(img)

```

```

def apply_gabor_stack(img, n_directions=8):

    """Apply Gabor filters in multiple orientations and combine
    responses."""

    responses = []

    for theta in np.linspace(0, np.pi, n_directions, endpoint=False):

        filt_real, filt_imag = gabor(img, frequency=0.1, theta=theta)

        responses.append(filt_real)

    combined = np.sum(responses, axis=0)

    combined = (combined - np.min(combined)) / (np.max(combined) -
    np.min(combined)) * 255

    return combined.astype(np.uint8)

```

```

def entropy_threshold(img):

    """Entropy-based threshold using Li's method (approx. entropy/Otsu
    hybrid)."""

    return threshold_li(img)

```

```

# Main Pipeline

def adaptive_optimized_vessel_detection(img_bgr):

    # [1] MSRPCP: Fix illumination/color balance

    img_corrected = MSRPCP(img_bgr)

    # [2] Extract green channel (most vessels visible)

    green = img_corrected[:, :, 1]

    # [3] Adaptive histogram equalization (contrast)

    adaptive_eq = adaptive_histogram_equalization(green)

    # [4] Gabor-based enhancement

```

```

gabor_enhanced = apply_gabor_stack(adaptive_eq)

# [5] Median filtering (remove impulsive noise)

median_filtered = cv2.medianBlur(gabor_enhanced, 3)

# [6] Thresholding (entropy / Li)

th_val = entropy_threshold(median_filtered)

vessel_mask = (median_filtered > th_val).astype(np.uint8) * 255

# [7] Morphological operations (refine vessels)

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3,3))

morph = cv2.morphologyEx(vessel_mask, cv2.MORPH_CLOSE, kernel,
iterations=2)

morph = cv2.morphologyEx(morph, cv2.MORPH_OPEN, kernel,
iterations=1)

# Cleanup small noise

morph_bool = morph > 0

morph_bool = remove_small_objects(morph_bool, min_size=100)

morph = morph_bool.astype(np.uint8) * 255

# [8] Contour detection + ROI cropping

contours, _ = cv2.findContours(morph, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

roi_images = []

for cnt in contours:

    x, y, w, h = cv2.boundingRect(cnt)

    if w > 10 and h > 10: # skip small artifacts

        roi = morph[y:y+h, x:x+w]

```

```

    roi_images.append(roi)

# ⑨ Feature scaling (for ML model input)

scaler = MinMaxScaler()

scaled_img = scaler.fit_transform(median_filtered)

scaled_img = (scaled_img * 255).astype(np.uint8)

return {
    "corrected": img_corrected,
    "green": green,
    "adaptive_eq": adaptive_eq,
    "gabor_enhanced": gabor_enhanced,
    "median_filtered": median_filtered,
    "vessel_mask": morph,
    "rois": roi_images,
    "scaled_for_ml": scaled_img
}

```

```

# Process All Images

for fname in os.listdir(img_dir):

    if fname.lower().endswith('.png', '.jpg', '.jpeg', '.tif'):

        path = os.path.join(img_dir, fname)

        img = cv2.imread(path)

        if img is None:

            continue

        results = adaptive_optimized_vessel_detection(img)

```

```
base = os.path.splitext(fname)[0]

cv2.imwrite(os.path.join(out_dir, f"{base}_corrected.png"),
cv2.cvtColor(results["corrected"], cv2.COLOR_RGB2BGR))

cv2.imwrite(os.path.join(out_dir, f"{base}_vessels.png"),
results["vessel_mask"])

cv2.imwrite(os.path.join(out_dir, f"{base}_scaled.png"),
results["scaled_for_ml"])

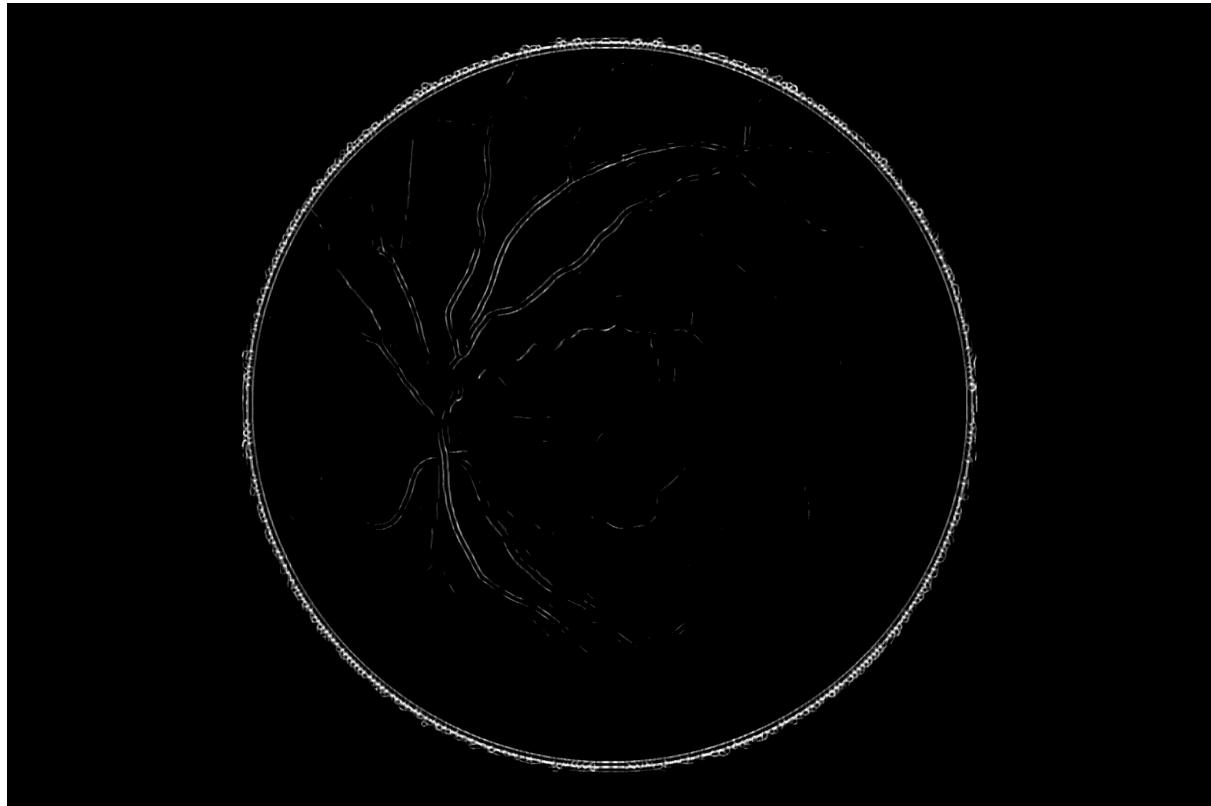
print(f"Processed: {fname}")

print("✅ Adaptive & Optimized Vessel Detection complete.")
```

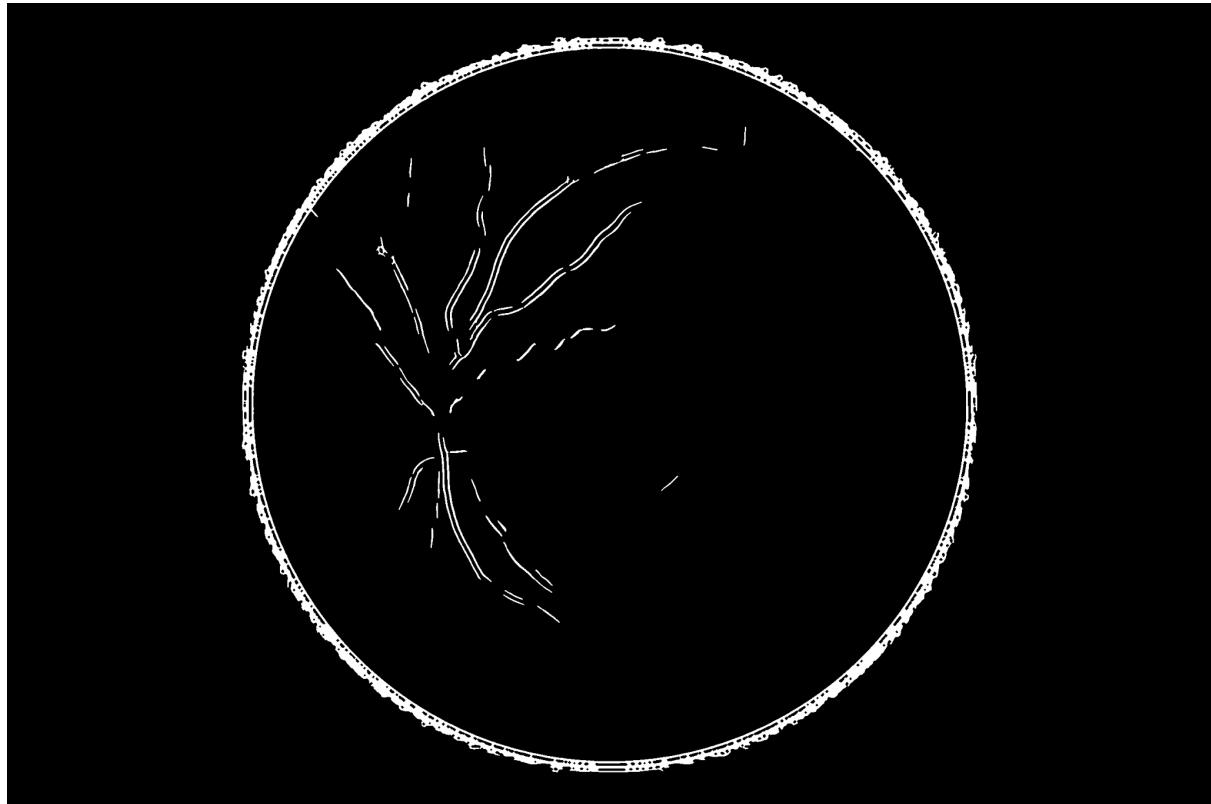
```
Processed: 0_left.jpg
Processed: 0_right.jpg
Processed: 10_left.jpg
Processed: 10_right.jpg
Processed: 11_left.jpg
Processed: 11_right.jpg
Processed: 12_left.jpg
Processed: 12_right.jpg
```



11\_left\_corrected.png



11\_left\_scaled.png



11\_left\_vessels.png

# Wavelet + Feature Scaling

## 2. RFI (Radio Frequency Interference) or Multi-scale Feature Extraction

Focus: texture, frequency, and multi-resolution analysis.

### **Pipeline C – Wavelet + Feature Scaling**

1. **Background removal** (normalize intensity map)
2. **Wavelet transform** (extract multi-scale RFI features)
3. **Optimal bandpass filter** (isolate signal frequency bands)
4. **Noise elimination**
5. **Normalization** (scale features)
6. **Thresholding (entropy or Otsu)** (extract interference zones)
7. **Contour detection + Shape cropping (ROI)**
8. **QRHMI** (radial invariant moment computation)
9. **Feature scaling**

 Use-case: Signal-image-based RFI detection with rotation-invariant texture features.

```
import cv2

import numpy as np

import os

import pywt

import matplotlib.pyplot as plt

from skimage.filters import threshold_li

from skimage.measure import moments_hu

from sklearn.preprocessing import StandardScaler

%matplotlib inline
```

```
# =====
```

```
# Directories
```

```
# =====
```

```
img_dir = "./datasets/training_images_limited"
```

```
out_dir = "./datasets/rfi_multiscale_feature_extraction"
```

```
os.makedirs(out_dir, exist_ok=True)
```

```
# =====
```

```
# Helper Functions
```

```
# =====
```

```
def normalize_intensity(img):
```

```
    """Normalize intensity map to [0,255]."""
```

```
    norm = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX)
```

```

    return norm.astype(np.uint8)

def apply_wavelet_transform(img, wavelet='db4', level=2):
    """
    Apply discrete wavelet transform and return a normalized
    multi-scale energy map.

    Automatically resizes sub-bands to match the approximation level.

    """
    img = img.astype(np.float32)

    coeffs = pywt.wavedec2(img, wavelet=wavelet, level=level)
    cA, cD = coeffs[0], coeffs[1:]

    energy = np.zeros_like(cA)

    for detail_level in cD:
        ch, cv_, cd_ = detail_level
        # Resize each to match cA dimensions
        ch_resized = cv2.resize(ch, (cA.shape[1], cA.shape[0]),
                               interpolation=cv2.INTER_LINEAR)
        cv_resized = cv2.resize(cv_, (cA.shape[1], cA.shape[0]),
                               interpolation=cv2.INTER_LINEAR)
        cd_resized = cv2.resize(cd_, (cA.shape[1], cA.shape[0]),
                               interpolation=cv2.INTER_LINEAR)

        energy += np.sqrt(ch_resized**2 + cv_resized**2 +
                          cd_resized**2)

    energy_norm = normalize_intensity(energy)
    return energy_norm

def bandpass_filter(img, low=0.01, high=0.2):

```

```

"""Apply FFT-based band-pass filter to isolate frequency band."""

rows, cols = img.shape

crow, ccol = rows // 2, cols // 2

mask = np.zeros((rows, cols), np.float32)

radius_low, radius_high = int(min(rows, cols)*low), int(min(rows,
cols)*high)

cv2.circle(mask, (ccol, crow), radius_high, 1, -1)
cv2.circle(mask, (ccol, crow), radius_low, 0, -1)

f = np.fft.fftshift(np.fft.fft2(img))

fshift = f * mask

filtered = np.abs(np.fft.ifft2(np.fft.ifftshift(fshift)))

return normalize_intensity(filtered)


def denoise(img, ksize=3):

    """Simple non-local or median filtering for noise removal."""

    return cv2.fastNlMeansDenoising(img, None, 10, 7, 21)


def threshold_entropy(img):

    """Entropy / Li threshold."""

    return threshold_li(img)


def compute_qrhmi(img_bin):

    """Compute quasi-radial / rotation-invariant Hu moments."""

    moments = moments_hu(img_bin)

    # Log transform for numerical stability

    with np.errstate(all='ignore'):

        log_moments = -np.sign(moments) * np.log10(np.abs(moments) +
1e-10)

```

```

    return log_moments

# =====
# Pipeline C: Wavelet + Feature Scaling
# =====

def rfi_multiscale_feature_extraction(img_gray):
    # [1] Background removal / normalization
    norm_img = normalize_intensity(img_gray)

    # [2] Wavelet transform - extract multi-scale texture features
    wavelet_features = apply_wavelet_transform(norm_img)

    # [3] Optimal band-pass filter (frequency isolation)
    bp_filtered = bandpass_filter(wavelet_features)

    # [4] Noise elimination
    denoised = denoise(bp_filtered)

    # [5] Normalization (feature scaling to [0,1])
    scaled = cv2.normalize(denoised.astype(np.float32), None, 0.0, 1.0,
                           cv2.NORM_MINMAX)

    # [6] Thresholding (entropy / Li)
    th_val = threshold_entropy((scaled * 255).astype(np.uint8))

    binary_mask = ((scaled * 255) > th_val).astype(np.uint8) * 255

```

```

# 7 Morphological cleanup

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3,3))

morph = cv2.morphologyEx(binary_mask, cv2.MORPH_OPEN, kernel,
iterations=2)

# 8 Contour detection + ROI cropping

contours, _ = cv2.findContours(morph, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

rois = []

for c in contours:

    x, y, w, h = cv2.boundingRect(c)

    if w > 10 and h > 10:

        roi = morph[y:y+h, x:x+w]

        rois.append(roi)

# 9 QRHMI (rotation-invariant moments of mask)

invariants = compute_qrhmi(morph)

# 10 Feature scaling (prepare descriptors for ML)

scaler = StandardScaler()

scaled_features =
scaler.fit_transform(invariants.reshape(-1,1)).flatten()

return {
    "normalized": norm_img,
    "wavelet": wavelet_features,
    "bandpass": bp_filtered,
    "denoised": denoised,
}

```

```
        "binary_mask": morph,
        "rois": rois,
        "qrhmi": scaled_features
    }

# =====
=====

# Batch Process All Images

#
=====

results_table = []

for fname in os.listdir(img_dir):
    if fname.lower().endswith(('.png', '.jpg', '.jpeg', '.tif')):
        path = os.path.join(img_dir, fname)
        img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
        if img is None:
            continue

        res = rfi_multiscale_feature_extraction(img)
        base = os.path.splitext(fname)[0]

        cv2.imwrite(os.path.join(out_dir, f"{base}_wavelet.png"),
                   res["wavelet"])

        cv2.imwrite(os.path.join(out_dir, f"{base}_rfi_mask.png"),
                   res["binary_mask"])

    results_table.append({}
```

```
        "filename": fname,
        "QRHMI_1": res["qrhmi"][0],
        "QRHMI_2": res["qrhmi"][1],
        "QRHMI_3": res["qrhmi"][2],
        "QRHMI_4": res["qrhmi"][3],
        "QRHMI_5": res["qrhmi"][4],
        "QRHMI_6": res["qrhmi"][5],
        "QRHMI_7": res["qrhmi"][6],
    }
print(f"Processed: {fname}")
```

```
import pandas as pd

pd.DataFrame(results_table).to_csv(os.path.join(out_dir,
"rfi_features.csv"), index=False)
```

```
print("✅ RFI / Multi-scale Feature Extraction completed.")
```

```
Processed: 0_left.jpg
Processed: 0_right.jpg
Processed: 10_left.jpg
Processed: 10_right.jpg
Processed: 11_left.jpg
Processed: 11_right.jpg
Processed: 12_left.jpg
Processed: 12_right.jpg
Processed: 13_left.jpg
Processed: 13_right.jpg
Processed: 14_left.jpg
```

Processed: 14\_right.jpg

Processed: 15\_left.jpg

Processed: 15\_right.jpg

Processed: 16\_left.jpg

Processed: 16\_right.jpg

Processed: 17\_left.jpg

Processed: 17\_right.jpg

Processed: 18\_left.jpg

Processed: 18\_right.jpg

Processed: 19\_left.jpg

Processed: 19\_right.jpg

Processed: 1\_left.jpg

Processed: 1\_right.jpg

Processed: 20\_left.jpg

Processed: 20\_right.jpg

Processed: 21\_left.jpg

Processed: 21\_right.jpg

Processed: 22\_left.jpg

Processed: 22\_right.jpg

Processed: 23\_left.jpg

Processed: 23\_right.jpg

Processed: 24\_left.jpg

Processed: 24\_right.jpg

Processed: 25\_left.jpg

Processed: 25\_right.jpg

Processed: 26\_left.jpg

Processed: 26\_right.jpg

Processed: 27\_left.jpg

Processed: 27\_right.jpg

Processed: 28\_left.jpg

Processed: 28\_right.jpg

Processed: 29\_left.jpg

Processed: 29\_right.jpg

Processed: 2\_left.jpg

Processed: 2\_right.jpg

Processed: 30\_left.jpg

Processed: 30\_right.jpg

Processed: 31\_left.jpg

Processed: 31\_right.jpg

Processed: 32\_left.jpg

Processed: 32\_right.jpg

Processed: 33\_left.jpg

Processed: 33\_right.jpg

Processed: 34\_left.jpg

Processed: 34\_right.jpg

Processed: 35\_left.jpg

Processed: 35\_right.jpg

Processed: 36\_left.jpg

Processed: 36\_right.jpg

Processed: 37\_left.jpg

Processed: 37\_right.jpg

Processed: 38\_left.jpg

Processed: 38\_right.jpg

Processed: 39\_left.jpg

Processed: 39\_right.jpg

Processed: 3\_left.jpg

Processed: 3\_right.jpg

Processed: 40\_left.jpg

Processed: 40\_right.jpg

Processed: 41\_left.jpg

Processed: 41\_right.jpg

Processed: 42\_left.jpg

Processed: 42\_right.jpg

Processed: 43\_left.jpg

Processed: 43\_right.jpg

Processed: 44\_left.jpg

Processed: 44\_right.jpg

Processed: 45\_left.jpg

Processed: 45\_right.jpg

Processed: 46\_left.jpg

Processed: 46\_right.jpg

Processed: 47\_left.jpg

Processed: 47\_right.jpg

Processed: 48\_left.jpg

Processed: 48\_right.jpg

Processed: 49\_left.jpg

Processed: 49\_right.jpg

Processed: 4\_left.jpg

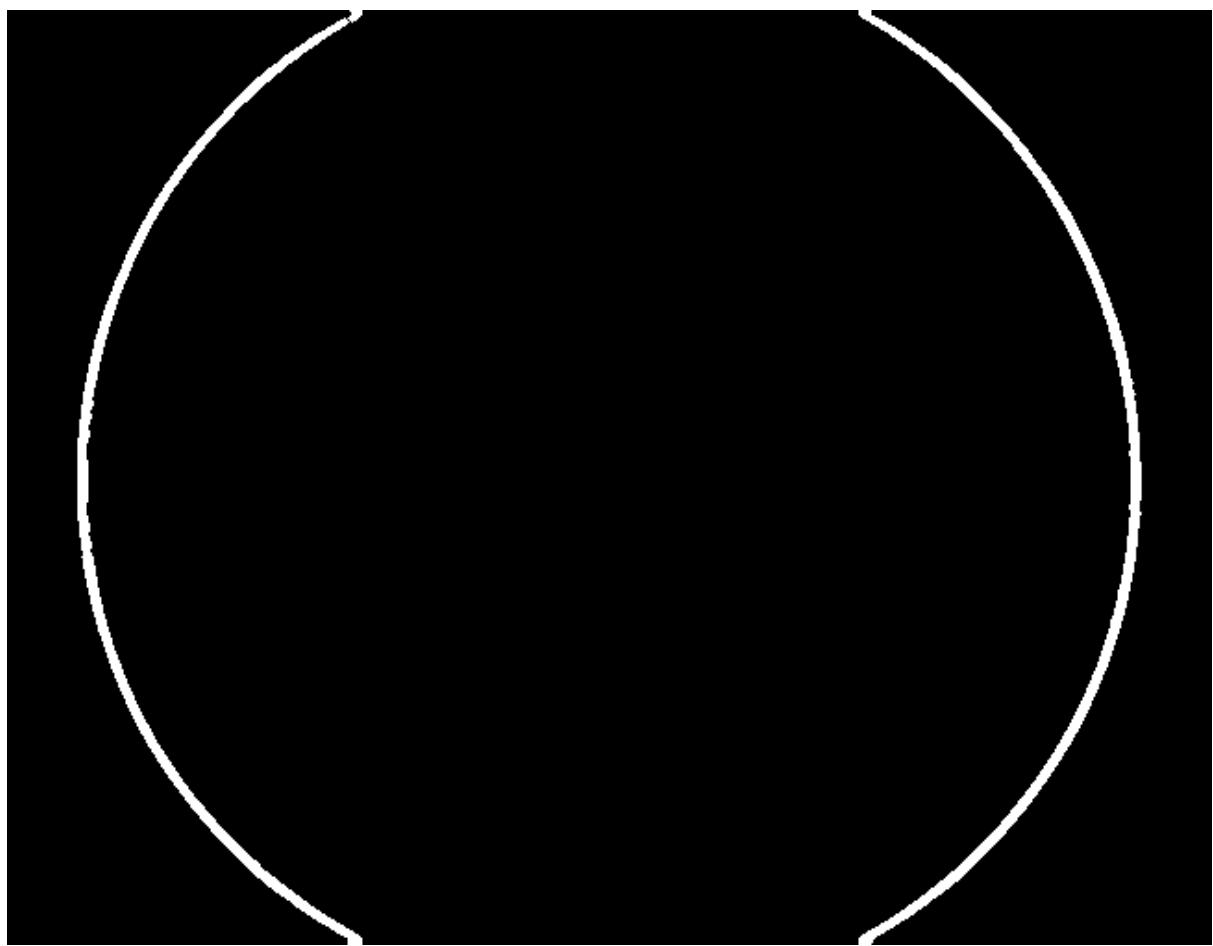
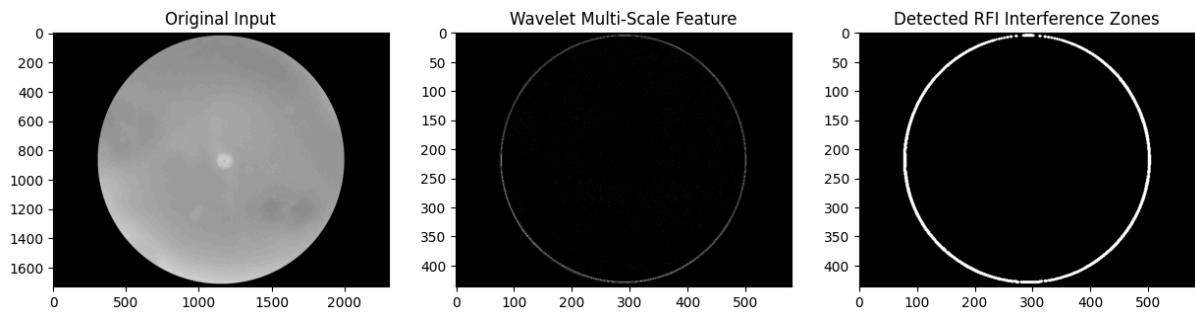
Processed: 4\_right.jpg

Processed: 5\_left.jpg

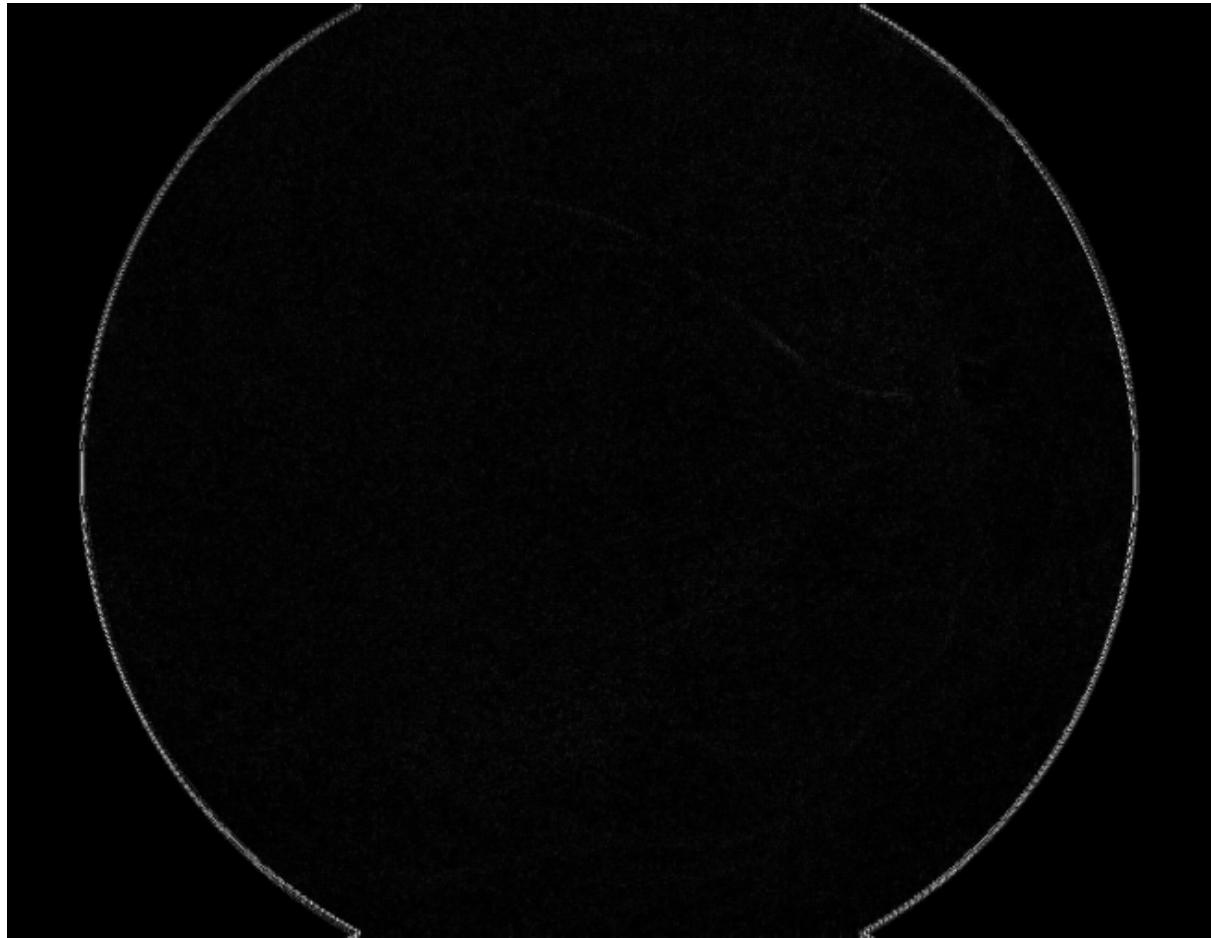
Processed: 5\_right.jpg

```
Processed: 6_left.jpg  
Processed: 6_right.jpg  
Processed: 7_left.jpg  
Processed: 7_right.jpg  
Processed: 8_left.jpg  
Processed: 8_right.jpg  
Processed: 9_left.jpg  
Processed: 9_right.jpg  
✓ RFI / Multi-scale Feature Extraction completed.
```

```
sample_name = os.listdir(img_dir)[0]  
base = os.path.splitext(sample_name)[0]  
  
original = cv2.imread(os.path.join(img_dir, sample_name),  
cv2.IMREAD_GRAYSCALE)  
  
wavelet = cv2.imread(os.path.join(out_dir, f"{base}_wavelet.png"),  
cv2.IMREAD_GRAYSCALE)  
  
mask = cv2.imread(os.path.join(out_dir, f"{base}_rfi_mask.png"),  
cv2.IMREAD_GRAYSCALE)  
  
plt.figure(figsize=(15,5))  
plt.subplot(1,3,1); plt.imshow(original, cmap='gray');  
plt.title('Original Input')  
  
plt.subplot(1,3,2); plt.imshow(wavelet, cmap='gray');  
plt.title('Wavelet Multi-Scale Feature')  
  
plt.subplot(1,3,3); plt.imshow(mask, cmap='gray'); plt.title('Detected  
RFI Interference Zones')  
plt.show()
```



10\_right\_rfi\_mask.png



10\_right\_wavelet.png

# Optimization-Aided RFI Feature Extraction

## **Pipeline D – Optimization-Aided RFI Feature Extraction**

1. **Wavelet transform**
2. **Optimal bandpass filter**
3. **Artificial Bee Colony (ABC)** optimization (auto-tune thresholds/filter parameters)
4. **Gabor-based filtering** (enhance texture features)
5. **Normalization**
6. **QRHMI** (rotation-invariant feature extraction)
7. **Feature scaling**

→ Use-case: Automated RFI or pattern classification with optimization-based feature tuning.

```

import cv2
import numpy as np
import os
import pywt
from skimage.filters import gabor
from skimage.measure import moments_hu
from sklearn.preprocessing import StandardScaler
from skimage.filters import threshold_li
from scipy.fft import fft2, ifft2, fftshift, ifftshift
import random
import matplotlib.pyplot as plt
%matplotlib inline

#
=====
=====
# Directories
#
=====

=====
img_dir = "./datasets/training_images_limited"
out_dir = "./datasets/optimization_aided_rfi_features"
os.makedirs(out_dir, exist_ok=True)

#
=====
=====
# Helper Functions
#
=====

=====
def normalize(img):
    """Normalize image to [0,255]."""
    norm = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX)
    return norm.astype(np.uint8)

def wavelet_transform(img, wavelet='db4', level=2):
    """Compute detail energy via wavelet decomposition."""
    img = img.astype(np.float32)
    coeffs = pywt.wavedec2(img, wavelet, level=level)
    cA, cD = coeffs[0], coeffs[1:]
    energy = np.zeros_like(cA)
    for ch, cv_, cd_ in cD:
        ch_res = cv2.resize(ch, (cA.shape[1], cA.shape[0]))
        cv_res = cv2.resize(cv_, (cA.shape[1], cA.shape[0]))
        cd_res = cv2.resize(cd_, (cA.shape[1], cA.shape[0]))
        energy += np.sqrt(ch_res**2 + cv_res**2 + cd_res**2)
    return normalize(energy)

def bandpass_filter(img, low=0.02, high=0.2):

```

```

"""FFT-based circular band-pass filter."""
rows, cols = img.shape
center = (rows//2, cols//2)
mask = np.zeros_like(img, dtype=np.float32)
r_low = int(min(rows, cols)*low)
r_high = int(min(rows, cols)*high)
cv2.circle(mask, center, r_high, 1, -1)
cv2.circle(mask, center, r_low, 0, -1)

f = fftshift(fft2(img))
fshift = f * mask
filtered = np.abs(ifft2(ifftshift(fshift)))
return normalize(filtered)

def gabor_enhancement(img, frequency=0.1, n_directions=8):
    """Aggregate Gabor filter responses for multiple orientations."""
    img = img.astype(np.float32)/255.0
    responses = []
    for theta in np.linspace(0, np.pi, n_directions, endpoint=False):
        real, _ = gabor(img, frequency=frequency, theta=theta)
        responses.append(real)
    combined = np.sum(responses, axis=0)
    return normalize(combined)

def compute_qrhmi(img_bin):
    """Compute rotation-invariant Hu Moments."""
    moments = moments_hu(img_bin)
    # Numerical stability with log-transform
    with np.errstate(all='ignore'):
        log_moments = -np.sign(moments) * np.log10(np.abs(moments)) +
1e-10
    return log_moments
# =====
# Artificial Bee Colony (ABC) Optimization - minimal implementation
#
=====

class ABC_Optimizer:
    """
    Simplified Artificial Bee Colony optimization to tune:
    [low_freq, high_freq, gabor_freq]
    Objective: maximize image edge/textured energy (variance).
    """
    def __init__(self, func, bounds, n_bees=10, n_iter=15):
        self.func = func
        self.bounds = np.array(bounds)
        self.n_bees = n_bees
        self.n_iter = n_iter

```

```

def optimize(self):
    # Initialize population randomly within bounds
    pop = [self.bounds[:,0] + np.random.rand(len(self.bounds)) * 
(self.bounds[:,1]-self.bounds[:,0])
            for _ in range(self.n_bees)]
    fitness = [self.func(x) for x in pop]

    for _ in range(self.n_iter):
        for i in range(self.n_bees):
            k = random.choice([idx for idx in range(self.n_bees) if
idx != i])
                phi = np.random.uniform(-1, 1, size=len(self.bounds))
                new_sol = pop[i] + phi * (pop[i] - pop[k])
                # Clip to bounds
                new_sol = np.clip(new_sol, self.bounds[:,0],
self.bounds[:,1])
                new_fit = self.func(new_sol)
                if new_fit > fitness[i]:
                    pop[i], fitness[i] = new_sol, new_fit
    best_index = np.argmax(fitness)
    return pop[best_index], fitness[best_index]

```

```

#
=====
=====

# Objective Function for ABC Optimization
#
=====

def objective_function(params, base_img):
    """Return variance (texture energy) of bandpass + gabor-enhanced
image."""
    low, high, gabor_freq = params
    # Ensure valid ordering
    if high <= low or low < 0.001:
        return 0
    bp = bandpass_filter(base_img, low, high)
    gabor_img = gabor_enhancement(bp, frequency=gabor_freq)
    return np.var(gabor_img)

#
=====
=====

# Main Pipeline
#
=====

def optimization_aided_rfi(img_gray):
    # Step 1: Wavelet transform (multi-scale base)

```

```

wave = wavelet_transform(img_gray)

# Step 2: ABC Optimization to auto-tune filter parameters
def fitness_func(p): return objective_function(p, wave)
abc = ABC_Optimizer(fitness_func, bounds=[(0.005,0.1), (0.1,0.4),
(0.05,0.25)], n_beans=10, n_iter=12)
best_params, score = abc.optimize()
low, high, gabor_freq = best_params
print(f"Optimized params: low={low:.4f}, high={high:.4f},
gabor_freq={gabor_freq:.4f}")

# Step 3: Apply tuned filtering sequence
bp = bandpass_filter(wave, low, high)
gabor_img = gabor_enhancement(bp, frequency=gabor_freq)

# Step 4: Normalization
norm = normalize(gabor_img)

# Step 5: Threshold + QRHMI
th = threshold_li(norm)
binary_mask = (norm > th).astype(np.uint8)*255
qrhmi_feats = compute_qrhmi(binary_mask)

# Step 6: Feature scaling (zero-mean, unit variance)
scaler = StandardScaler()
scaled_feats =
scaler.fit_transform(qrhmi_feats.reshape(-1,1)).flatten()

return {
    "wavelet": wave,
    "bandpass": bp,
    "gabor_optimized": gabor_img,
    "binary_mask": binary_mask,
    "qrhmi_scaled": scaled_feats,
    "best_params": best_params
}

# =====
# Batch Run on Folder
#
=====

results_table = []
for fname in os.listdir(img_dir):
    if fname.lower().endswith('.png','.jpg','.jpeg','.tif'):
        path = os.path.join(img_dir,fname)
        img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
        if img is None:
            continue

```

```
res = optimization_aided_rfi(img)
base = os.path.splitext(fname)[0]
cv2.imwrite(os.path.join(out_dir, f"{base}_gabor_opt.png"),
res["gabor_optimized"])
cv2.imwrite(os.path.join(out_dir, f"{base}_mask.png"),
res["binary_mask"])

results_table.append({
    "filename": fname,
    "param_low": res["best_params"][0],
    "param_high": res["best_params"][1],
    "param_gaborfreq": res["best_params"][2],
    **{f"QRHMI_{i+1}": val for i, val in
enumerate(res["qrhmi_scaled"])}}
)
print(f"✓ Processed {fname}")

import pandas as pd
pd.DataFrame(results_table).to_csv(os.path.join(out_dir,
"optimized_rfi_features.csv"), index=False)
print("✓ Optimization-Aided RFI Feature Extraction complete.")
```

Output: Took infinite time to preprocess.

# General Enhancement + Normalization



### 3. General Image Enhancement / Denoising

Focus: removing artifacts, enhancing contrast, preparing clean input.

#### **Pipeline E – General Enhancement + Normalization**

1. **Background removal / Mask application**
2. **Noise elimination / Median filtering**
3. **Gaussian filtering**
4. **CLAHE** (local contrast improvement)
5. **Gamma correction**
6. **Normalization**
7. **Morphological operations**
8. **Artifact removal**
9. **Image resizing**

→ Use-case: Preprocessing medical or photographic images before segmentation/classification.

```

import cv2

import numpy as np

import os

import matplotlib.pyplot as plt

%matplotlib inline


# =====
=====

# Directories

# =====
=====

img_dir = "./datasets/training_images_limited/"

out_dir = "./datasets/general_enhancement_normalization"

os.makedirs(out_dir, exist_ok=True)


# =====
=====

# Helper Functions

# =====
=====

def apply_mask(img, mask=None) :

    """Apply a binary mask to remove background."""

    if mask is None:

        # Create a simple mask via intensity threshold (auto)

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        _, mask = cv2.threshold(gray, 10, 255, cv2.THRESH_BINARY)

    masked = cv2.bitwise_and(img, img, mask=mask)

```

```

    return masked

def median_filter(img, ksize=3):

    """Median filtering to remove impulsive / salt-and-pepper
noise."""

    return cv2.medianBlur(img, ksize)

def gaussian_filter(img, ksize=5, sigma=0):

    """Gaussian smoothing."""

    return cv2.GaussianBlur(img, (ksize, ksize), sigma)

def clahe_enhancement(img):

    """Apply CLAHE on each channel for local contrast improvement."""

    if len(img.shape) == 2:

        clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))

        return clahe.apply(img)

    else:

        lab = cv2.cvtColor(img, cv2.COLOR_BGR2LAB)

        l, a, b = cv2.split(lab)

        clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))

        l_clahe = clahe.apply(l)

        lab = cv2.merge((l_clahe, a, b))

    return cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)

def gamma_correction(img, gamma=1.1):

    """Adjust brightness via gamma correction."""

    inv_gamma = 1.0 / gamma

    table = np.array([(i / 255.0) ** inv_gamma * 255

```

```

        for i in np.arange(256) ])).astype("uint8")

    return cv2.LUT(img, table)

def normalize_intensity(img):
    """Normalize intensity values to [0,255]."""

    norm = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX)

    return norm.astype(np.uint8)

def morphological_cleanup(img):
    """Morphological open/close to refine structures."""

    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3,3))

    opened = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)

    closed = cv2.morphologyEx(opened, cv2.MORPH_CLOSE, kernel)

    return closed

def remove_small_artifacts(binary, min_size=100):
    """Remove small disconnected components."""

    num_labels, labels, stats, _ =
    cv2.connectedComponentsWithStats(binary, connectivity=8)

    cleaned = np.zeros_like(binary)

    for i in range(1, num_labels):

        if stats[i, cv2.CC_STAT_AREA] >= min_size:

            cleaned[labels == i] = 255

    return cleaned

def resize_image(img, target_size=(512, 512)):

    """Resize image to target shape."""

    return cv2.resize(img, target_size, interpolation=cv2.INTER_AREA)

```

```

#
=====

# Full Enhancement + Normalization Pipeline

#
=====

=====

def general_enhancement_normalization(img_bgr, mask=None,
target_size=(512, 512)):

    # [1] Background removal

    masked = apply_mask(img_bgr, mask)

    # [2] Noise elimination (median)

    medianed = median_filter(masked, 3)

    # [3] Gaussian smoothing

    smoothed = gaussian_filter(medianed, 5, 0)

    # [4] CLAHE (local contrast improvement)

    clahe_img = clahe_enhancement(smoothed)

    # [5] Gamma correction

    gamma_corrected = gamma_correction(clahe_img, gamma=1.2)

    # [6] Normalization

    normalized = normalize_intensity(gamma_corrected)

    # [7] Morphological operations

    morph = morphological_cleanup(normalized)

```

```

# [8] Artifact removal (optional for maskd-based cleanup)

gray = cv2.cvtColor(morph, cv2.COLOR_BGR2GRAY)

_, binary = cv2.threshold(gray, 10, 255, cv2.THRESH_BINARY)

cleaned_mask = remove_small_artifacts(binary, min_size=200)

final = cv2.bitwise_and(morph, morph, mask=cleaned_mask)

# [9] Resizing

resized = resize_image(final, target_size)

return resized

# =====

# Batch Process

# =====

for fname in os.listdir(img_dir):

    if fname.lower().endswith(('.png', '.jpg', '.jpeg', '.tif')):

        path = os.path.join(img_dir, fname)

        img = cv2.imread(path)

        if img is None:

            continue

            result = general_enhancement_normalization(img)

            base = os.path.splitext(fname)[0]

            cv2.imwrite(os.path.join(out_dir, f'{base}_enhanced.png'), result)

```

```
print(f"Processed: {fname}")

print("✅ General Enhancement + Normalization completed.")

sample_name = os.listdir(img_dir)[0]
base = os.path.splitext(sample_name)[0]

original = cv2.imread(os.path.join(img_dir, sample_name))
enhanced = cv2.imread(os.path.join(out_dir, f"{base}_enhanced.png"))

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1); plt.imshow(cv2.cvtColor(original,
cv2.COLOR_BGR2RGB)); plt.title("Original Image")

plt.subplot(1, 2, 2); plt.imshow(cv2.cvtColor(enhanced,
cv2.COLOR_BGR2RGB)); plt.title("Enhanced & Normalized")

plt.show()
```

```
Processed: 0_left.jpg
```

```
Processed: 0_right.jpg
```

```
Processed: 10_left.jpg
```

```
Processed: 10_right.jpg
```

```
Processed: 11_left.jpg
```

```
Processed: 11_right.jpg
```

```
Processed: 12_left.jpg
```

```
Processed: 12_right.jpg
```

```
Processed: 13_left.jpg
```

```
Processed: 13_right.jpg
```

```
Processed: 14_left.jpg
```

Processed: 14\_right.jpg

Processed: 15\_left.jpg

Processed: 15\_right.jpg

Processed: 16\_left.jpg

Processed: 16\_right.jpg

Processed: 17\_left.jpg

Processed: 17\_right.jpg

Processed: 18\_left.jpg

Processed: 18\_right.jpg

Processed: 19\_left.jpg

Processed: 19\_right.jpg

Processed: 1\_left.jpg

Processed: 1\_right.jpg

Processed: 20\_left.jpg

Processed: 20\_right.jpg

Processed: 21\_left.jpg

Processed: 21\_right.jpg

Processed: 22\_left.jpg

Processed: 22\_right.jpg

Processed: 23\_left.jpg

Processed: 23\_right.jpg

Processed: 24\_left.jpg

Processed: 24\_right.jpg

Processed: 25\_left.jpg

Processed: 25\_right.jpg

Processed: 26\_left.jpg

Processed: 26\_right.jpg

Processed: 27\_left.jpg

Processed: 27\_right.jpg

Processed: 28\_left.jpg

Processed: 28\_right.jpg

Processed: 29\_left.jpg

Processed: 29\_right.jpg

Processed: 2\_left.jpg

Processed: 2\_right.jpg

Processed: 30\_left.jpg

Processed: 30\_right.jpg

Processed: 31\_left.jpg

Processed: 31\_right.jpg

Processed: 32\_left.jpg

Processed: 32\_right.jpg

Processed: 33\_left.jpg

Processed: 33\_right.jpg

Processed: 34\_left.jpg

Processed: 34\_right.jpg

Processed: 35\_left.jpg

Processed: 35\_right.jpg

Processed: 36\_left.jpg

Processed: 36\_right.jpg

Processed: 37\_left.jpg

Processed: 37\_right.jpg

Processed: 38\_left.jpg

Processed: 38\_right.jpg

Processed: 39\_left.jpg

Processed: 39\_right.jpg

Processed: 3\_left.jpg

Processed: 3\_right.jpg

Processed: 40\_left.jpg

Processed: 40\_right.jpg

Processed: 41\_left.jpg

Processed: 41\_right.jpg

Processed: 42\_left.jpg

Processed: 42\_right.jpg

Processed: 43\_left.jpg

Processed: 43\_right.jpg

Processed: 44\_left.jpg

Processed: 44\_right.jpg

Processed: 45\_left.jpg

Processed: 45\_right.jpg

Processed: 46\_left.jpg

Processed: 46\_right.jpg

Processed: 47\_left.jpg

Processed: 47\_right.jpg

Processed: 48\_left.jpg

Processed: 48\_right.jpg

Processed: 49\_left.jpg

Processed: 49\_right.jpg

Processed: 4\_left.jpg

Processed: 4\_right.jpg

Processed: 5\_left.jpg

Processed: 5\_right.jpg

Processed: 6\_left.jpg

Processed: 6\_right.jpg

Processed: 7\_left.jpg

Processed: 7\_right.jpg

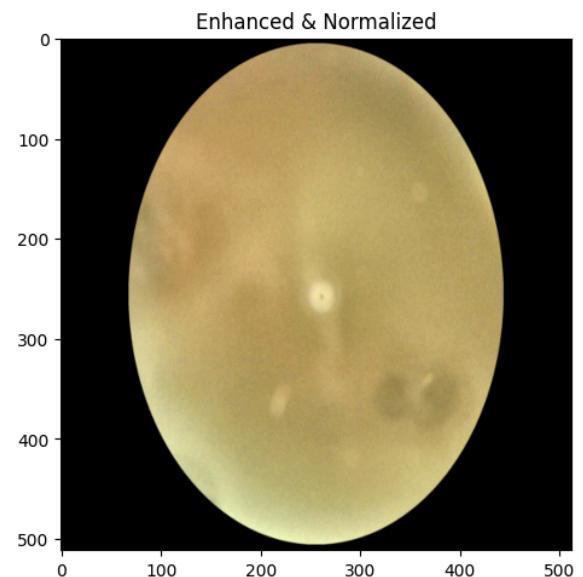
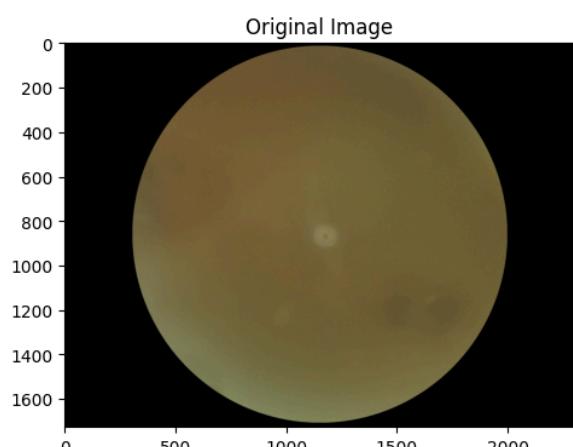
Processed: 8\_left.jpg

Processed: 8\_right.jpg

Processed: 9\_left.jpg

Processed: 9\_right.jpg

✓ General Enhancement + Normalization completed.





2\_left\_enhanced.png



2\_right\_enhanced.png

**odir\_clahe\_top**

```

import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D,
Dropout
from tensorflow.keras.optimizers import Adam
import pandas as pd
import os
from sklearn.model_selection import train_test_split # <-- IMPORT THIS

# --- 1. Define Constants ---
TRAIN_DIR = 'final_kals_mz/final_dataset/Training_Images_clahe_tophat/'
# TEST_DIR is not used in this version, as we validate from the
training set
# TEST_DIR = 'final_kals_mz/final_dataset/Testing_Images_clahe_tophat/'
CSV_PATH = 'full_df.csv'

# Model parameters
IMG_SIZE = (299, 299)
BATCH_SIZE = 32
EPOCHS = 15
LEARNING_RATE = 0.001
VALIDATION_SPLIT = 0.2 # Use 20% of the data for validation

# --- 2. Load and Prepare DataFrame ---
try:
    df = pd.read_csv(CSV_PATH)
except FileNotFoundError:
    print(f"Error: Cannot find the CSV file at {CSV_PATH}")
    exit()

df['class_label'] = df['labels'].str.extract(r"\'(.*?)\'")
df.dropna(subset=['class_label'], inplace=True)

print(f"Loaded {len(df)} total records from {CSV_PATH}")

# --- NEW: Split the DataFrame ---
train_df, validation_df = train_test_split(
    df,
    test_size=VALIDATION_SPLIT,
    random_state=42,
    stratify=df['class_label'] # Ensures both sets have similar class
distribution
)

print(f"Using {len(train_df)} images for training.")
print(f"Using {len(validation_df)} images for validation.")
print("-" * 30)

```

```

# --- 3. Create In-Memory Data Augmentation and Generators ---
train_datagen = ImageDataGenerator(
    preprocessing_function=tf.keras.applications.inception_v3.preprocess_input,
    rotation_range=30,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

# For the validation set, we only do the required preprocessing
validation_datagen = ImageDataGenerator(
    preprocessing_function=tf.keras.applications.inception_v3.preprocess_input
)

# --- 4. Create Data Generators from DataFrame ---
print("Creating Training Generator...")
train_generator = train_datagen.flow_from_dataframe(
    dataframe=train_df, # <-- USE train_df
    directory=TRAIN_DIR,
    x_col='filename',
    y_col='class_label',
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=True
)

print("\nCreating Validation Generator...")
validation_generator = validation_datagen.flow_from_dataframe(
    dataframe=validation_df, # <-- USE validation_df
    directory=TRAIN_DIR,      # <-- Still reads from TRAIN_DIR
    x_col='filename',
    y_col='class_label',
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=False
)

# --- FIXED: Get the number of classes ---
num_classes = len(train_generator.class_indices)
print(f"\nModel will be trained on {num_classes} classes.")
print("Class Indices:", train_generator.class_indices)
print("-" * 30)

```

```

# --- 5. Build the Transfer Learning Model ---
print("Building InceptionV3 model with transfer learning...")

base_model = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(IMG_SIZE[0], IMG_SIZE[1], 3)
)
base_model.trainable = False

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(num_classes, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

# --- 6. Compile the Model ---
model.compile(
    optimizer=Adam(learning_rate=LEARNING_RATE),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
model.summary()
print("-" * 30)

# --- 7. Train the Model ---
print(f"Starting model training for {EPOCHS} epochs...")

history = model.fit(
    train_generator,
    epochs=EPOCHS,
    validation_data=validation_generator, # <-- USE
    validation_generator
    steps_per_epoch=train_generator.samples // BATCH_SIZE,
    validation_steps=validation_generator.samples // BATCH_SIZE,
    verbose=1
)

print("Training complete.")
print("-" * 30)

# --- 8. Evaluate the Model on Validation Set ---
print("Evaluating model on the validation set...")
results = model.evaluate(validation_generator)
print(f"Validation Loss: {results[0]:.4f}")
print(f"Validation Accuracy: {results[1]:.4f}")
print("-" * 30)

```

```

# --- 9. Save the Model ---
MODEL_SAVE_PATH = 'fundus_inceptionv3_model.keras'
model.save(MODEL_SAVE_PATH)
print(f"Model successfully saved to: {MODEL_SAVE_PATH}")

Error: Cannot find the CSV file at full_df.csv
Loaded 6392 total records from full_df.csv
Using 5113 images for training.
Using 1279 images for validation.
-----
Creating Training Generator...
Found 5113 validated image filenames belonging to 8 classes.

Creating Validation Generator...
Found 1279 validated image filenames belonging to 8 classes.

Model will be trained on 8 classes.
Class Indices: {'A': 0, 'C': 1, 'D': 2, 'G': 3, 'H': 4, 'M': 5, 'N': 6,
'O': 7}
-----
Building InceptionV3 model with transfer learning...
Downloading data from
https://storage.googleapis.com/tensorflow/keras-applications/inception_
v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 18s 0us/step

```

**Model: "functional"**

| Layer (type)                         | Output Shape         | Param # | Connected to |
|--------------------------------------|----------------------|---------|--------------|
| input_layer<br>(InputLayer)          | (None, 299, 299, 3)  | 0       | -            |
| conv2d (Conv2D)<br>input_layer[0][0] | (None, 149, 149, 32) | 864     | -            |
| batch_normalization                  | (None, 149, 149, 96) | 96      | conv2d[0][0] |

|                      |                       |                  |  |        |                |
|----------------------|-----------------------|------------------|--|--------|----------------|
|                      | (BatchNormalizatio... | 32)              |  |        |                |
|                      | activation            | (None, 149, 149, |  | 0      |                |
| batch_normalizat...  |                       |                  |  |        |                |
|                      | (Activation)          | 32)              |  |        |                |
|                      |                       |                  |  |        |                |
|                      | conv2d_1 (Conv2D)     | (None, 147, 147, |  | 9,216  |                |
| activation[0][0]     |                       |                  |  |        |                |
|                      |                       | 32)              |  |        |                |
|                      |                       |                  |  |        |                |
|                      | batch_normalizatio... | (None, 147, 147, |  | 96     | conv2d_1[0][0] |
|                      | (BatchNormalizatio... | 32)              |  |        |                |
|                      |                       |                  |  |        |                |
|                      | activation_1          | (None, 147, 147, |  | 0      |                |
| batch_normalizat...  |                       |                  |  |        |                |
|                      | (Activation)          | 32)              |  |        |                |
|                      |                       |                  |  |        |                |
|                      | conv2d_2 (Conv2D)     | (None, 147, 147, |  | 18,432 |                |
| activation_1[0][...] |                       |                  |  |        |                |
|                      |                       | 64)              |  |        |                |
|                      |                       |                  |  |        |                |
|                      | batch_normalizatio... | (None, 147, 147, |  | 192    | conv2d_2[0][0] |
|                      | (BatchNormalizatio... | 64)              |  |        |                |
|                      |                       |                  |  |        |                |
|                      | activation_2          | (None, 147, 147, |  | 0      |                |
| batch_normalizat...  |                       |                  |  |        |                |
|                      | (Activation)          | 64)              |  |        |                |
|                      |                       |                  |  |        |                |
|                      | max_pooling2d         | (None, 73, 73,   |  | 0      |                |
| activation_2[0][...] |                       |                  |  |        |                |
|                      | (MaxPooling2D)        | 64)              |  |        |                |
|                      |                       |                  |  |        |                |

|   |                        |  |         |                |
|---|------------------------|--|---------|----------------|
|   |                        |  |         |                |
| conv2d_3 (Conv2D)<br>max_pooling2d[0]...            | (None, 73, 73,<br>80)  |  | 5,120   |                |
|   |                        |  |         |                |
| batch_normalizatio...<br>(BatchNormalizatio...      | (None, 73, 73,<br>80)  |  | 240     | conv2d_3[0][0] |
|   |                        |  |         |                |
| activation_3<br>batch_normalizat...<br>(Activation) | (None, 73, 73,<br>80)  |  | 0       |                |
|   |                        |  |         |                |
| conv2d_4 (Conv2D)<br>activation_3[0] [...           | (None, 71, 71,<br>192) |  | 138,240 |                |
|   |                        |  |         |                |
| batch_normalizatio...<br>(BatchNormalizatio...      | (None, 71, 71,<br>192) |  | 576     | conv2d_4[0][0] |
|   |                        |  |         |                |
| activation_4<br>batch_normalizat...<br>(Activation) | (None, 71, 71,<br>192) |  | 0       |                |
|   |                        |  |         |                |
| max_pooling2d_1<br>activation_4[0] [...             | (None, 35, 35,<br>192) |  | 0       |                |
|   |                        |  |         |                |
| conv2d_8 (Conv2D)<br>max_pooling2d_1[...            | (None, 35, 35,<br>64)  |  | 12,288  |                |
|   |                        |  |         |                |

|                       |                |  |        |                |
|-----------------------|----------------|--|--------|----------------|
| batch_normalizatio... | (None, 35, 35, |  | 192    | conv2d_8[0][0] |
| (BatchNormalizatio... | 64)            |  |        |                |
| activation_8          | (None, 35, 35, |  | 0      |                |
| batch_normalizat...   |                |  |        |                |
| (Activation)          | 64)            |  |        |                |
| conv2d_6 (Conv2D)     | (None, 35, 35, |  | 9,216  |                |
| max_pooling2d_1 [...] | 48)            |  |        |                |
| conv2d_9 (Conv2D)     | (None, 35, 35, |  | 55,296 |                |
| activation_8[0] [...] | 96)            |  |        |                |
| batch_normalizatio... | (None, 35, 35, |  | 144    | conv2d_6[0][0] |
| (BatchNormalizatio... | 48)            |  |        |                |
| batch_normalizatio... | (None, 35, 35, |  | 288    | conv2d_9[0][0] |
| (BatchNormalizatio... | 96)            |  |        |                |
| activation_6          | (None, 35, 35, |  | 0      |                |
| batch_normalizat...   |                |  |        |                |
| (Activation)          | 48)            |  |        |                |
| activation_9          | (None, 35, 35, |  | 0      |                |
| batch_normalizat...   |                |  |        |                |
| (Activation)          | 96)            |  |        |                |
| average_pooling2d     | (None, 35, 35, |  | 0      |                |
| max_pooling2d_1 [...] |                |  |        |                |

|   |                       |        |                |
|---|-----------------------|--------|----------------|
| (AveragePooling2D)  | (192)                 |        |                |
| conv2d_5 (Conv2D)<br>max_pooling2d_1[...]                       | (None, 35, 35,<br>64) | 12,288 |                |
| conv2d_7 (Conv2D)<br>activation_6[0] [...]                      | (None, 35, 35,<br>64) | 76,800 |                |
| conv2d_10 (Conv2D)<br>activation_9[0] [...]                     | (None, 35, 35,<br>96) | 82,944 |                |
| conv2d_11 (Conv2D)<br>average_pooling2...                       | (None, 35, 35,<br>32) | 6,144  |                |
| batch_normalizatio...   (BatchNormalizatio...                   | (None, 35, 35,<br>64) | 192    | conv2d_5[0][0] |
| batch_normalizatio...   (BatchNormalizatio...                   | (None, 35, 35,<br>64) | 192    | conv2d_7[0][0] |
| batch_normalizatio...   conv2d_10[0][0]   (BatchNormalizatio... | (None, 35, 35,<br>96) | 288    |                |
| batch_normalizatio...   conv2d_11[0][0]   (BatchNormalizatio... | (None, 35, 35,<br>32) | 96     |                |

|   |                        |        |     |              |
|---|------------------------|--------|-----|--------------|
|   |                        |        |     |              |
| activation_5<br>batch_normalizat...<br>(Activation)   | (None, 35, 35,<br>64)  |        | 0   |              |
|   |                        |        |     |              |
| activation_7<br>batch_normalizat...<br>(Activation)   | (None, 35, 35,<br>64)  |        | 0   |              |
|   |                        |        |     |              |
| activation_10<br>batch_normalizat...<br>(Activation)  | (None, 35, 35,<br>96)  |        | 0   |              |
|   |                        |        |     |              |
| activation_11<br>batch_normalizat...<br>(Activation)  | (None, 35, 35,<br>32)  |        | 0   |              |
|   |                        |        |     |              |
| mixed0<br>activation_5[0] [...<br>(Concatenate)<br>activation_7[0] [...<br>activation_10[0]...<br>activation_11[0]...<br>activation_15 (Conv2D) | (None, 35, 35,<br>256) |        | 0   |              |
|   |                        |        |     |              |
| batch_normalizatio...<br>conv2d_15[0][0]<br>(BatchNormalizatio...<br>activation_15<br>batch_normalizat...                                       | (None, 35, 35,<br>64)  | 16,384 | 192 | mixed0[0][0] |
|   |                        |        |     |              |

|  |   |                        |        |                |  |
|--|---|------------------------|--------|----------------|--|
|  | (Activation)  | 64)                    |        |                |  |
|  | conv2d_13 (Conv2D)  | (None, 35, 35,<br>48)  | 12,288 | mixed0 [0] [0] |  |
|  | conv2d_16 (Conv2D)<br>activation_15[0]...                               | (None, 35, 35,<br>96)  | 55,296 |                |  |
|  | batch_normalizatio...<br>conv2d_13[0][0]<br>(BatchNormalizatio...   48) | (None, 35, 35,         | 144    |                |  |
|  | batch_normalizatio...<br>conv2d_16[0][0]<br>(BatchNormalizatio...   96) | (None, 35, 35,         | 288    |                |  |
|  | activation_13<br>batch_normalizat...<br>(Activation)   48)              | (None, 35, 35,         | 0      |                |  |
|  | activation_16<br>batch_normalizat...<br>(Activation)   96)              | (None, 35, 35,         | 0      |                |  |
|  | average_pooling2d_1<br>(AveragePooling2D)   256)                        | (None, 35, 35,<br>256) | 0      | mixed0 [0] [0] |  |
|  | conv2d_12 (Conv2D)  | (None, 35, 35,<br>64)  | 16,384 | mixed0 [0] [0] |  |

|  |                       |  |        |
|--|-----------------------|--|--------|
|  |                       |  |        |
| conv2d_14 (Conv2D)<br>activation_13[0]...                                | (None, 35, 35,<br>64) |  | 76,800 |
|  |                       |  |        |
| conv2d_17 (Conv2D)<br>activation_16[0]...                                | (None, 35, 35,<br>96) |  | 82,944 |
|  |                       |  |        |
| conv2d_18 (Conv2D)<br>average_pooling2...<br>average_pooling2...         | (None, 35, 35,<br>64) |  | 16,384 |
|  |                       |  |        |
| batch_normalizatio...<br>conv2d_12[0][0]<br>(BatchNormalizatio...<br>64) | (None, 35, 35,<br>64) |  | 192    |
|  |                       |  |        |
| batch_normalizatio...<br>conv2d_14[0][0]<br>(BatchNormalizatio...<br>64) | (None, 35, 35,<br>64) |  | 192    |
|  |                       |  |        |
| batch_normalizatio...<br>conv2d_17[0][0]<br>(BatchNormalizatio...<br>96) | (None, 35, 35,<br>96) |  | 288    |
|  |                       |  |        |
| batch_normalizatio...<br>conv2d_18[0][0]<br>(BatchNormalizatio...<br>64) | (None, 35, 35,<br>64) |  | 192    |
|  |                       |  |        |
| activation_12<br>batch_normalizat...<br>(Activation)                     | (None, 35, 35,<br>64) |  | 0      |
|  |                       |  |        |

|                       |                |        |              |
|-----------------------|----------------|--------|--------------|
| activation_14         | (None, 35, 35, | 0      |              |
| batch_normalizat...   |                |        |              |
| (Activation)          | 64)            |        |              |
| activation_17         | (None, 35, 35, | 0      |              |
| batch_normalizat...   |                |        |              |
| (Activation)          | 96)            |        |              |
| activation_18         | (None, 35, 35, | 0      |              |
| batch_normalizat...   |                |        |              |
| (Activation)          | 64)            |        |              |
| mixed1                | (None, 35, 35, | 0      |              |
| activation_12[0]...   |                |        |              |
| (Concatenate)         | 288)           |        |              |
| activation_14[0]...   |                |        |              |
| activation_17[0]...   |                |        |              |
| activation_18[0]...   |                |        |              |
| conv2d_22 (Conv2D)    | (None, 35, 35, | 18,432 | mixed1[0][0] |
|                       | 64)            |        |              |
| batch_normalizatio... | (None, 35, 35, | 192    |              |
| conv2d_22[0][0]       |                |        |              |
| (BatchNormalizatio... | 64)            |        |              |
| activation_22         | (None, 35, 35, | 0      |              |
| batch_normalizat...   |                |        |              |
| (Activation)          | 64)            |        |              |
| conv2d_20 (Conv2D)    | (None, 35, 35, | 13,824 | mixed1[0][0] |
|                       | 48)            |        |              |

|  |                        |  |        |              |
|--|------------------------|--|--------|--------------|
|  |                        |  |        |              |
| conv2d_23 (Conv2D)<br>activation_22[0]...                                | (None, 35, 35,<br>96)  |  | 55,296 |              |
|  |                        |  |        |              |
| batch_normalizatio...<br>conv2d_20[0][0]<br>(BatchNormalizatio...<br>48) | (None, 35, 35,<br>48)  |  | 144    |              |
|  |                        |  |        |              |
| batch_normalizatio...<br>conv2d_23[0][0]<br>(BatchNormalizatio...<br>96) | (None, 35, 35,<br>96)  |  | 288    |              |
|  |                        |  |        |              |
| activation_20<br>batch_normalizat...<br>(Activation)<br>48)              | (None, 35, 35,<br>48)  |  | 0      |              |
|  |                        |  |        |              |
| activation_23<br>batch_normalizat...<br>(Activation)<br>96)              | (None, 35, 35,<br>96)  |  | 0      |              |
|  |                        |  |        |              |
| average_pooling2d_2<br>(AveragePooling2D)                                | (None, 35, 35,<br>288) |  | 0      | mixed1[0][0] |
|  |                        |  |        |              |
| conv2d_19 (Conv2D)<br>64)  | (None, 35, 35,<br>64)  |  | 18,432 | mixed1[0][0] |
|  |                        |  |        |              |
| conv2d_21 (Conv2D)<br>activation_20[0]...<br>64)                         | (None, 35, 35,<br>64)  |  | 76,800 |              |
|  |                        |  |        |              |

|                       |                       |        |
|-----------------------|-----------------------|--------|
| conv2d_24 (Conv2D)    | (None, 35, 35,        | 82,944 |
| activation_23[0]...   | 96)                   |        |
| conv2d_25 (Conv2D)    | (None, 35, 35,        | 18,432 |
| average_pooling2...   | 64)                   |        |
| batch_normalizatio... | (None, 35, 35,        | 192    |
| conv2d_19[0][0]       | (BatchNormalizatio... |        |
|                       | 64)                   |        |
| batch_normalizatio... | (None, 35, 35,        | 192    |
| conv2d_21[0][0]       | (BatchNormalizatio... |        |
|                       | 64)                   |        |
| batch_normalizatio... | (None, 35, 35,        | 288    |
| conv2d_24[0][0]       | (BatchNormalizatio... |        |
|                       | 96)                   |        |
| batch_normalizatio... | (None, 35, 35,        | 192    |
| conv2d_25[0][0]       | (BatchNormalizatio... |        |
|                       | 64)                   |        |
| activation_19         | (None, 35, 35,        | 0      |
| batch_normalizat...   | (Activation)          |        |
|                       | 64)                   |        |
| activation_21         | (None, 35, 35,        | 0      |
| batch_normalizat...   | (Activation)          |        |
|                       | 64)                   |        |
| activation_24         | (None, 35, 35,        | 0      |
| batch_normalizat...   |                       |        |

|                       |                       |                |        |              |
|-----------------------|-----------------------|----------------|--------|--------------|
|                       | (Activation)          | 96             |        |              |
|                       |                       |                |        |              |
| activation_25         | batch_normalizat...   | (None, 35, 35, | 0      |              |
|                       | (Activation)          | 64)            |        |              |
|                       |                       |                |        |              |
| mixed2                | activation_19[0]...   | (None, 35, 35, | 0      |              |
|                       | (Concatenate)         | 288)           |        |              |
| activation_21[0]...   |                       |                |        |              |
|                       |                       |                |        |              |
| activation_24[0]...   |                       |                |        |              |
|                       |                       |                |        |              |
| activation_25[0]...   |                       |                |        |              |
|                       |                       |                |        |              |
| conv2d_27 (Conv2D)    |                       | (None, 35, 35, | 18,432 | mixed2[0][0] |
|                       |                       | 64)            |        |              |
|                       |                       |                |        |              |
| batch_normalizatio... | conv2d_27[0][0]       | (None, 35, 35, | 192    |              |
|                       | (BatchNormalizatio... | 64)            |        |              |
|                       |                       |                |        |              |
| activation_27         | batch_normalizat...   | (None, 35, 35, | 0      |              |
|                       | (Activation)          | 64)            |        |              |
|                       |                       |                |        |              |
| conv2d_28 (Conv2D)    | activation_27[0]...   | (None, 35, 35, | 55,296 |              |
|                       |                       | 96)            |        |              |
|                       |                       |                |        |              |
| batch_normalizatio... | conv2d_28[0][0]       | (None, 35, 35, | 288    |              |
|                       | (BatchNormalizatio... | 96)            |        |              |
|                       |                       |                |        |              |

|                       |                |         |               |
|-----------------------|----------------|---------|---------------|
| activation_28         | (None, 35, 35, | 0       |               |
| batch_normalizat...   |                |         |               |
| (Activation)          | 96)            |         |               |
|                       |                |         |               |
| conv2d_26 (Conv2D)    | (None, 17, 17, | 995,328 | mixed2[0] [0] |
|                       | 384)           |         |               |
|                       |                |         |               |
| conv2d_29 (Conv2D)    | (None, 17, 17, | 82,944  |               |
| activation_28[0]...   | 96)            |         |               |
|                       |                |         |               |
| batch_normalizatio... | (None, 17, 17, | 1,152   |               |
| conv2d_26[0] [0]      |                |         |               |
| (BatchNormalizatio... | 384)           |         |               |
|                       |                |         |               |
| batch_normalizatio... | (None, 17, 17, | 288     |               |
| conv2d_29[0] [0]      |                |         |               |
| (BatchNormalizatio... | 96)            |         |               |
|                       |                |         |               |
| activation_26         | (None, 17, 17, | 0       |               |
| batch_normalizat...   |                |         |               |
| (Activation)          | 384)           |         |               |
|                       |                |         |               |
| activation_29         | (None, 17, 17, | 0       |               |
| batch_normalizat...   |                |         |               |
| (Activation)          | 96)            |         |               |
|                       |                |         |               |
| max_pooling2d_2       | (None, 17, 17, | 0       | mixed2[0] [0] |
|                       |                |         |               |
| (MaxPooling2D)        | 288)           |         |               |
|                       |                |         |               |
| mixed3                | (None, 17, 17, | 0       |               |
| activation_26[0]...   |                |         |               |

|                     |                       |                          |              |  |
|---------------------|-----------------------|--------------------------|--------------|--|
|                     | (Concatenate)         | (None, 768)              |              |  |
| activation_29[0]... |                       |                          |              |  |
|                     | max_pooling2d_2[...]  |                          |              |  |
|                     | conv2d_34 (Conv2D)    | (None, 17, 17, 98, 304)  | mixed3[0][0] |  |
|                     |                       | 128)                     |              |  |
|                     | batch_normalizatio... | (None, 17, 17, 384)      |              |  |
| conv2d_34[0][0]     |                       |                          |              |  |
|                     | (BatchNormalizatio... | 128)                     |              |  |
|                     | activation_34         | (None, 17, 17, 0)        |              |  |
| batch_normalizat... |                       |                          |              |  |
|                     | (Activation)          | 128)                     |              |  |
|                     | conv2d_35 (Conv2D)    | (None, 17, 17, 114, 688) |              |  |
| activation_34[0]... |                       |                          |              |  |
|                     |                       | 128)                     |              |  |
|                     | batch_normalizatio... | (None, 17, 17, 384)      |              |  |
| conv2d_35[0][0]     |                       |                          |              |  |
|                     | (BatchNormalizatio... | 128)                     |              |  |
|                     | activation_35         | (None, 17, 17, 0)        |              |  |
| batch_normalizat... |                       |                          |              |  |
|                     | (Activation)          | 128)                     |              |  |
|                     | conv2d_31 (Conv2D)    | (None, 17, 17, 98, 304)  | mixed3[0][0] |  |
|                     |                       | 128)                     |              |  |
|                     | conv2d_36 (Conv2D)    | (None, 17, 17, 114, 688) |              |  |
| activation_35[0]... |                       |                          |              |  |

|                       |                       |      |         |  |
|-----------------------|-----------------------|------|---------|--|
|                       |                       | 128) |         |  |
|                       |                       |      |         |  |
| batch_normalizatio... | (None, 17, 17,        |      | 384     |  |
| conv2d_31[0][0]       | (BatchNormalizatio... | 128) |         |  |
|                       |                       |      |         |  |
| batch_normalizatio... | (None, 17, 17,        |      | 384     |  |
| conv2d_36[0][0]       | (BatchNormalizatio... | 128) |         |  |
|                       |                       |      |         |  |
| activation_31         | (None, 17, 17,        |      | 0       |  |
| batch_normalizat...   | (Activation)          | 128) |         |  |
|                       |                       |      |         |  |
| activation_36         | (None, 17, 17,        |      | 0       |  |
| batch_normalizat...   | (Activation)          | 128) |         |  |
|                       |                       |      |         |  |
| conv2d_32 (Conv2D)    | (None, 17, 17,        |      | 114,688 |  |
| activation_31[0]...   | 128)                  |      |         |  |
|                       |                       |      |         |  |
| conv2d_37 (Conv2D)    | (None, 17, 17,        |      | 114,688 |  |
| activation_36[0]...   | 128)                  |      |         |  |
|                       |                       |      |         |  |
| batch_normalizatio... | (None, 17, 17,        |      | 384     |  |
| conv2d_32[0][0]       | (BatchNormalizatio... | 128) |         |  |
|                       |                       |      |         |  |
| batch_normalizatio... | (None, 17, 17,        |      | 384     |  |
| conv2d_37[0][0]       | (BatchNormalizatio... | 128) |         |  |
|                       |                       |      |         |  |

|  |                        |         |               |               |  |
|--|------------------------|---------|---------------|---------------|--|
|  |                        |         |               |               |  |
| activation_32<br>batch_normalizat...<br>(Activation)               | (None, 17, 17,<br>128) |         | 0             |               |  |
|  |                        |         |               |               |  |
| activation_37<br>batch_normalizat...<br>(Activation)               | (None, 17, 17,<br>128) |         | 0             |               |  |
|  |                        |         |               |               |  |
| average_pooling2d_3<br>(AveragePooling2D)                          | (None, 17, 17,<br>768) |         | 0             | mixed3[0] [0] |  |
|  |                        |         |               |               |  |
| conv2d_30 (Conv2D)   | (None, 17, 17,<br>192) | 147,456 | mixed3[0] [0] |               |  |
|  |                        |         |               |               |  |
| conv2d_33 (Conv2D)<br>activation_32[0]...<br>(Activation)          | (None, 17, 17,<br>192) | 172,032 |               |               |  |
|  |                        |         |               |               |  |
| conv2d_38 (Conv2D)<br>activation_37[0]...<br>(Activation)          | (None, 17, 17,<br>192) | 172,032 |               |               |  |
|  |                        |         |               |               |  |
| conv2d_39 (Conv2D)<br>average_pooling2...<br>(AveragePooling2D)    | (None, 17, 17,<br>192) | 147,456 |               |               |  |
|  |                        |         |               |               |  |
| batch_normalizatio...<br>conv2d_30[0][0]<br>(BatchNormalizatio...) | (None, 17, 17,<br>192) | 576     |               |               |  |
|  |                        |         |               |               |  |

|                              |                      |                     |     |
|------------------------------|----------------------|---------------------|-----|
| batch_normalization_33[0][0] | (BatchNormalization) | (None, 17, 17, 192) | 576 |
| conv2d_33[0][0]              |                      |                     |     |
| batch_normalization_38[0][0] | (BatchNormalization) | (None, 17, 17, 192) | 576 |
| conv2d_38[0][0]              |                      |                     |     |
| batch_normalization_39[0][0] | (BatchNormalization) | (None, 17, 17, 192) | 576 |
| conv2d_39[0][0]              |                      |                     |     |
| activation_30                | (Activation)         | (None, 17, 17, 192) | 0   |
| batch_normalizat...          |                      |                     |     |
| activation_33                | (Activation)         | (None, 17, 17, 192) | 0   |
| batch_normalizat...          |                      |                     |     |
| activation_38                | (Activation)         | (None, 17, 17, 192) | 0   |
| batch_normalizat...          |                      |                     |     |
| activation_39                | (Activation)         | (None, 17, 17, 192) | 0   |
| batch_normalizat...          |                      |                     |     |
| mixed4                       | (Concatenate)        | (None, 17, 17, 768) | 0   |
| activation_30[0]...          |                      |                     |     |
| activation_33[0]...          |                      |                     |     |
| activation_38[0]...          |                      |                     |     |
| activation_39[0]...          |                      |                     |     |

|   |                        |  |         |              |
|---|------------------------|--|---------|--------------|
|   |                        |  |         |              |
| conv2d_44 (Conv2D)  | (None, 17, 17,<br>160) |  | 122,880 | mixed4[0][0] |
| batch_normalizatio...<br>conv2d_44[0][0]<br>(BatchNormalizatio...)  | (None, 17, 17,<br>160) |  | 480     |              |
| activation_44<br>batch_normalizat...<br>(Activation)                | (None, 17, 17,<br>160) |  | 0       |              |
| conv2d_45 (Conv2D)<br>activation_44[0]...<br>(None, 17, 17,<br>160) | (None, 17, 17,<br>160) |  | 179,200 |              |
| batch_normalizatio...<br>conv2d_45[0][0]<br>(BatchNormalizatio...)  | (None, 17, 17,<br>160) |  | 480     |              |
| activation_45<br>batch_normalizat...<br>(Activation)                | (None, 17, 17,<br>160) |  | 0       |              |
| conv2d_41 (Conv2D)  | (None, 17, 17,<br>160) |  | 122,880 | mixed4[0][0] |
| conv2d_46 (Conv2D)<br>activation_45[0]...<br>(None, 17, 17,<br>160) | (None, 17, 17,<br>160) |  | 179,200 |              |
|   |                        |  |         |              |

|                        |                              |  |
|------------------------|------------------------------|--|
| batch_normalization_41 | (None, 17, 17, 480)          |  |
| conv2d_41[0][0]        | (BatchNormalization)   160   |  |
|                        |                              |  |
| batch_normalization_46 | (None, 17, 17, 480)          |  |
| conv2d_46[0][0]        | (BatchNormalization)   160   |  |
|                        |                              |  |
| activation_41          | (None, 17, 17, 0)            |  |
| batch_normalizat...    | (Activation)   160           |  |
|                        |                              |  |
| activation_46          | (None, 17, 17, 0)            |  |
| batch_normalizat...    | (Activation)   160           |  |
|                        |                              |  |
| conv2d_42 (Conv2D)     | (None, 17, 17, 179,200)      |  |
| activation_41[0]...    | 160                          |  |
|                        |                              |  |
| conv2d_47 (Conv2D)     | (None, 17, 17, 179,200)      |  |
| activation_46[0]...    | 160                          |  |
|                        |                              |  |
| batch_normalizatio...  | (None, 17, 17, 480)          |  |
| conv2d_42[0][0]        | (BatchNormalizatio...   160) |  |
|                        |                              |  |
| batch_normalizatio...  | (None, 17, 17, 480)          |  |
| conv2d_47[0][0]        | (BatchNormalizatio...   160) |  |
|                        |                              |  |
| activation_42          | (None, 17, 17, 0)            |  |
| batch_normalizat...    |                              |  |

|  |   |                          |  |                |
|--|---|--------------------------|--|----------------|
|  | (Activation)                              | (None, 17, 17, 160)      |  |                |
|  | activation_47<br>batch_normalizat...      | (None, 17, 17, 0)        |  |                |
|  | (Activation)                              | (None, 17, 160)          |  |                |
|  | average_pooling2d_4<br>(AveragePooling2D) | (None, 17, 17, 0)        |  | mixed4 [0] [0] |
|  | conv2d_40 (Conv2D)                        | (None, 17, 17, 147, 456) |  | mixed4 [0] [0] |
|  |   | (None, 17, 17, 192)      |  |                |
|  | conv2d_43 (Conv2D)<br>activation_42[0]... | (None, 17, 17, 215, 040) |  |                |
|  |   | (None, 17, 17, 192)      |  |                |
|  | conv2d_48 (Conv2D)<br>activation_47[0]... | (None, 17, 17, 215, 040) |  |                |
|  |   | (None, 17, 17, 192)      |  |                |
|  | conv2d_49 (Conv2D)<br>average_pooling2... | (None, 17, 17, 147, 456) |  |                |
|  |   | (None, 17, 17, 192)      |  |                |
|  | batch_normalizatio...<br>conv2d_40[0][0]  | (None, 17, 17, 576)      |  |                |
|  | (BatchNormalizatio...   192)              |                          |  |                |
|  | batch_normalizatio...<br>conv2d_43[0][0]  | (None, 17, 17, 576)      |  |                |
|  | (BatchNormalizatio...   192)              |                          |  |                |

|                       |                       |      |         |              |
|-----------------------|-----------------------|------|---------|--------------|
|                       |                       |      |         |              |
| batch_normalizatio... | (None, 17, 17,        |      | 576     |              |
| conv2d_48[0][0]       | (BatchNormalizatio... | 192) |         |              |
|                       |                       |      |         |              |
| batch_normalizatio... | (None, 17, 17,        |      | 576     |              |
| conv2d_49[0][0]       | (BatchNormalizatio... | 192) |         |              |
|                       |                       |      |         |              |
| activation_40         | (None, 17, 17,        |      | 0       |              |
| batch_normalizat...   | (Activation)          | 192) |         |              |
|                       |                       |      |         |              |
| activation_43         | (None, 17, 17,        |      | 0       |              |
| batch_normalizat...   | (Activation)          | 192) |         |              |
|                       |                       |      |         |              |
| activation_48         | (None, 17, 17,        |      | 0       |              |
| batch_normalizat...   | (Activation)          | 192) |         |              |
|                       |                       |      |         |              |
| activation_49         | (None, 17, 17,        |      | 0       |              |
| batch_normalizat...   | (Activation)          | 192) |         |              |
|                       |                       |      |         |              |
| mixed5                | (None, 17, 17,        |      | 0       |              |
| activation_40[0]...   | (Concatenate)         | 768) |         |              |
| activation_43[0]...   |                       |      |         |              |
| activation_48[0]...   |                       |      |         |              |
| activation_49[0]...   |                       |      |         |              |
|                       |                       |      |         |              |
| conv2d_54 (Conv2D)    | (None, 17, 17,        |      | 122,880 | mixed5[0][0] |

|                       |                       |      |         |              |
|-----------------------|-----------------------|------|---------|--------------|
|                       |                       | 160) |         |              |
|                       |                       |      |         |              |
| batch_normalizatio... | (None, 17, 17,        |      | 480     |              |
| conv2d_54[0][0]       | (BatchNormalizatio... | 160) |         |              |
|                       |                       |      |         |              |
| activation_54         | (None, 17, 17,        |      | 0       |              |
| batch_normalizat...   | (Activation)          | 160) |         |              |
|                       |                       |      |         |              |
| conv2d_55 (Conv2D)    | (None, 17, 17,        |      | 179,200 |              |
| activation_54[0]...   | 160)                  |      |         |              |
|                       |                       |      |         |              |
| batch_normalizatio... | (None, 17, 17,        |      | 480     |              |
| conv2d_55[0][0]       | (BatchNormalizatio... | 160) |         |              |
|                       |                       |      |         |              |
| activation_55         | (None, 17, 17,        |      | 0       |              |
| batch_normalizat...   | (Activation)          | 160) |         |              |
|                       |                       |      |         |              |
| conv2d_51 (Conv2D)    | (None, 17, 17,        |      | 122,880 | mixed5[0][0] |
|                       | 160)                  |      |         |              |
|                       |                       |      |         |              |
| conv2d_56 (Conv2D)    | (None, 17, 17,        |      | 179,200 |              |
| activation_55[0]...   | 160)                  |      |         |              |
|                       |                       |      |         |              |
| batch_normalizatio... | (None, 17, 17,        |      | 480     |              |
| conv2d_51[0][0]       | (BatchNormalizatio... | 160) |         |              |

|                       |                       |         |  |
|-----------------------|-----------------------|---------|--|
|                       |                       |         |  |
| batch_normalizatio... | (None, 17, 17,        | 480     |  |
| conv2d_56[0][0]       | (BatchNormalizatio... |         |  |
|                       | 160)                  |         |  |
|                       |                       |         |  |
| activation_51         | (None, 17, 17,        | 0       |  |
| batch_norm...         | (Activation)          |         |  |
|                       | 160)                  |         |  |
|                       |                       |         |  |
| activation_56         | (None, 17, 17,        | 0       |  |
| batch_norm...         | (Activation)          |         |  |
|                       | 160)                  |         |  |
|                       |                       |         |  |
| conv2d_52 (Conv2D)    | (None, 17, 17,        | 179,200 |  |
| activation_51[0]...   | 160)                  |         |  |
|                       |                       |         |  |
| conv2d_57 (Conv2D)    | (None, 17, 17,        | 179,200 |  |
| activation_56[0]...   | 160)                  |         |  |
|                       |                       |         |  |
| batch_normalizatio... | (None, 17, 17,        | 480     |  |
| conv2d_52[0][0]       | (BatchNormalizatio... |         |  |
|                       | 160)                  |         |  |
|                       |                       |         |  |
| batch_normalizatio... | (None, 17, 17,        | 480     |  |
| conv2d_57[0][0]       | (BatchNormalizatio... |         |  |
|                       | 160)                  |         |  |
|                       |                       |         |  |
| activation_52         | (None, 17, 17,        | 0       |  |
| batch_norm...         | (Activation)          |         |  |
|                       | 160)                  |         |  |
|                       |                       |         |  |

|                       |                |         |              |
|-----------------------|----------------|---------|--------------|
| activation_57         | (None, 17, 17, | 0       |              |
| batch_normalizat...   |                |         |              |
| (Activation)          | 160            |         |              |
|                       |                |         |              |
| average_pooling2d_5   | (None, 17, 17, | 0       | mixed5[0][0] |
| (AveragePooling2D)    | 768            |         |              |
|                       |                |         |              |
| conv2d_50 (Conv2D)    | (None, 17, 17, | 147,456 | mixed5[0][0] |
| 192)                  |                |         |              |
|                       |                |         |              |
| conv2d_53 (Conv2D)    | (None, 17, 17, | 215,040 |              |
| activation_52[0]...   | 192)           |         |              |
|                       |                |         |              |
| conv2d_58 (Conv2D)    | (None, 17, 17, | 215,040 |              |
| activation_57[0]...   | 192)           |         |              |
|                       |                |         |              |
| conv2d_59 (Conv2D)    | (None, 17, 17, | 147,456 |              |
| average_pooling2...   | 192)           |         |              |
|                       |                |         |              |
| batch_normalizatio... | (None, 17, 17, | 576     |              |
| conv2d_50[0][0]       |                |         |              |
| (BatchNormalizatio... | 192)           |         |              |
|                       |                |         |              |
| batch_normalizatio... | (None, 17, 17, | 576     |              |
| conv2d_53[0][0]       |                |         |              |
| (BatchNormalizatio... | 192)           |         |              |
|                       |                |         |              |
| batch_normalizatio... | (None, 17, 17, | 576     |              |
| conv2d_58[0][0]       |                |         |              |

|                     |                       |                |         |              |
|---------------------|-----------------------|----------------|---------|--------------|
|                     | (BatchNormalizatio... | 192)           |         |              |
|                     | batch_normalizatio... | (None, 17, 17, | 576     |              |
| conv2d_59[0][0]     |                       |                |         |              |
|                     | (BatchNormalizatio... | 192)           |         |              |
|                     | activation_50         | (None, 17, 17, | 0       |              |
| batch_normalizat... |                       |                |         |              |
|                     | (Activation)          | 192)           |         |              |
|                     | activation_53         | (None, 17, 17, | 0       |              |
| batch_normalizat... |                       |                |         |              |
|                     | (Activation)          | 192)           |         |              |
|                     | activation_58         | (None, 17, 17, | 0       |              |
| batch_normalizat... |                       |                |         |              |
|                     | (Activation)          | 192)           |         |              |
|                     | activation_59         | (None, 17, 17, | 0       |              |
| batch_normalizat... |                       |                |         |              |
|                     | (Activation)          | 192)           |         |              |
|                     | mixed6                | (None, 17, 17, | 0       |              |
| activation_50[0]... |                       |                |         |              |
|                     | (Concatenate)         | 768)           |         |              |
| activation_53[0]... |                       |                |         |              |
|                     | activation_58[0]...   |                |         |              |
|                     | activation_59[0]...   |                |         |              |
|                     | conv2d_64 (Conv2D)    | (None, 17, 17, | 147,456 | mixed6[0][0] |
|                     |                       | 192)           |         |              |

|                                |                      |                         |         |                |
|--------------------------------|----------------------|-------------------------|---------|----------------|
| batch_normalization_64 [0] [0] | (BatchNormalization) | (None, 17, 17, 192)     | 576     |                |
| activation_64                  | (Activation)         | (None, 17, 17, 192)     | 0       |                |
| conv2d_65 (Conv2D)             | (Conv2D)             | (None, 17, 17, 258,048) | 258,048 |                |
| activation_64 [0] ...          |                      | (192)                   |         |                |
| batch_normalization_65 [0] [0] | (BatchNormalization) | (None, 17, 17, 192)     | 576     |                |
| activation_65                  | (Activation)         | (None, 17, 17, 192)     | 0       |                |
| conv2d_61 (Conv2D)             | (Conv2D)             | (None, 17, 17, 147,456) | 147,456 | mixed6 [0] [0] |
|                                |                      | (192)                   |         |                |
| conv2d_66 (Conv2D)             | (Conv2D)             | (None, 17, 17, 258,048) | 258,048 |                |
| activation_65 [0] ...          |                      | (192)                   |         |                |
| batch_normalization_61 [0] [0] | (BatchNormalization) | (None, 17, 17, 192)     | 576     |                |
| batch_normalization_66 [0] [0] | (BatchNormalization) | (None, 17, 17, 192)     | 576     |                |

|   |                              |  |  |  |
|---|------------------------------|--|--|--|
|   | (BatchNormalizatio...   192) |  |  |  |
|   |                              |  |  |  |
| activation_61<br>batch_normalizat...   (Activation) | (None, 17, 17,   0           |  |  |  |
|   | 192)                         |  |  |  |
|   |                              |  |  |  |
| activation_66<br>batch_normalizat...   (Activation) | (None, 17, 17,   0           |  |  |  |
|   | 192)                         |  |  |  |
|   |                              |  |  |  |
| conv2d_62 (Conv2D)<br>activation_61[0]...           | (None, 17, 17,   258,048     |  |  |  |
|   | 192)                         |  |  |  |
|   |                              |  |  |  |
| conv2d_67 (Conv2D)<br>activation_66[0]...           | (None, 17, 17,   258,048     |  |  |  |
|   | 192)                         |  |  |  |
|   |                              |  |  |  |
| batch_normalizatio...   (None, 17, 17,   576        |                              |  |  |  |
| conv2d_62[0][0]   (BatchNormalizatio...   192)      |                              |  |  |  |
|   |                              |  |  |  |
| batch_normalizatio...   (None, 17, 17,   576        |                              |  |  |  |
| conv2d_67[0][0]   (BatchNormalizatio...   192)      |                              |  |  |  |
|   |                              |  |  |  |
| activation_62<br>batch_normalizat...   (Activation) | (None, 17, 17,   0           |  |  |  |
|   | 192)                         |  |  |  |
|   |                              |  |  |  |
| activation_67<br>batch_normalizat...   (Activation) | (None, 17, 17,   0           |  |  |  |
|   | 192)                         |  |  |  |

|  |                        |  |         |               |  |
|--|------------------------|--|---------|---------------|--|
|  |                        |  |         |               |  |
| average_pooling2d_6<br>(AveragePooling2D)                                | (None, 17, 17,<br>768) |  | 0       | mixed6[0] [0] |  |
| conv2d_60 (Conv2D)   | (None, 17, 17,<br>192) |  | 147,456 | mixed6[0] [0] |  |
| conv2d_63 (Conv2D)<br>activation_62[0]...                                | (None, 17, 17,<br>192) |  | 258,048 |               |  |
| conv2d_68 (Conv2D)<br>activation_67[0]...                                | (None, 17, 17,<br>192) |  | 258,048 |               |  |
| conv2d_69 (Conv2D)<br>average_pooling2...                                | (None, 17, 17,<br>192) |  | 147,456 |               |  |
| batch_normalizatio...<br>conv2d_60[0][0]<br>(BatchNormalizatio...   192) | (None, 17, 17,<br>192) |  | 576     |               |  |
| batch_normalizatio...<br>conv2d_63[0][0]<br>(BatchNormalizatio...   192) | (None, 17, 17,<br>192) |  | 576     |               |  |
| batch_normalizatio...<br>conv2d_68[0][0]<br>(BatchNormalizatio...   192) | (None, 17, 17,<br>192) |  | 576     |               |  |
|  |                        |  |         |               |  |

|                       |                |         |              |  |
|-----------------------|----------------|---------|--------------|--|
|                       |                |         |              |  |
| batch_normalizatio... | (None, 17, 17, |         | 576          |  |
| conv2d_69[0][0]       |                |         |              |  |
| (BatchNormalizatio... | 192)           |         |              |  |
|                       |                |         |              |  |
| activation_60         | (None, 17, 17, |         | 0            |  |
| batch_normalizat...   |                |         |              |  |
| (Activation)          | 192)           |         |              |  |
|                       |                |         |              |  |
| activation_63         | (None, 17, 17, |         | 0            |  |
| batch_normalizat...   |                |         |              |  |
| (Activation)          | 192)           |         |              |  |
|                       |                |         |              |  |
| activation_68         | (None, 17, 17, |         | 0            |  |
| batch_normalizat...   |                |         |              |  |
| (Activation)          | 192)           |         |              |  |
|                       |                |         |              |  |
| activation_69         | (None, 17, 17, |         | 0            |  |
| batch_normalizat...   |                |         |              |  |
| (Activation)          | 192)           |         |              |  |
|                       |                |         |              |  |
| mixed7                | (None, 17, 17, |         | 0            |  |
| activation_60[0]...   |                |         |              |  |
| (Concatenate)         | 768)           |         |              |  |
| activation_63[0]...   |                |         |              |  |
|                       |                |         |              |  |
| activation_68[0]...   |                |         |              |  |
|                       |                |         |              |  |
| activation_69[0]...   |                |         |              |  |
|                       |                |         |              |  |
| conv2d_72 (Conv2D)    | (None, 17, 17, | 147,456 | mixed7[0][0] |  |
|                       | 192)           |         |              |  |
|                       |                |         |              |  |
| batch_normalizatio... | (None, 17, 17, |         | 576          |  |
| conv2d_72[0][0]       |                |         |              |  |
| (BatchNormalizatio... | 192)           |         |              |  |
|                       |                |         |              |  |

|  |                            |              |   |  |  |
|--|----------------------------|--------------|---|--|--|
|  |                            |              |   |  |  |
| activation_72<br>batch_normalizat...<br>(Activation)                     | (None, 17, 17,<br>192)     |              | 0 |  |  |
|  |                            |              |   |  |  |
| conv2d_73 (Conv2D)<br>activation_72[0]...<br>  192)                      | (None, 17, 17,<br>258,048) |              |   |  |  |
|  |                            |              |   |  |  |
| batch_normalizatio...<br>conv2d_73[0][0]<br>(BatchNormalizatio...   192) | (None, 17, 17,<br>576)     |              |   |  |  |
|  |                            |              |   |  |  |
| activation_73<br>batch_normalizat...<br>(Activation)                     | (None, 17, 17,<br>192)     |              | 0 |  |  |
|  |                            |              |   |  |  |
| conv2d_70 (Conv2D)<br>  192)   | (None, 17, 17,<br>147,456) | mixed7[0][0] |   |  |  |
|  |                            |              |   |  |  |
| conv2d_74 (Conv2D)<br>activation_73[0]...<br>  192)                      | (None, 17, 17,<br>258,048) |              |   |  |  |
|  |                            |              |   |  |  |
| batch_normalizatio...<br>conv2d_70[0][0]<br>(BatchNormalizatio...   192) | (None, 17, 17,<br>576)     |              |   |  |  |
|  |                            |              |   |  |  |
| batch_normalizatio...<br>conv2d_74[0][0]<br>(BatchNormalizatio...   192) | (None, 17, 17,<br>576)     |              |   |  |  |
|  |                            |              |   |  |  |

|                       |                   |         |              |
|-----------------------|-------------------|---------|--------------|
| activation_70         | (None, 17, 17,    | 0       |              |
| batch_normalizat...   |                   |         |              |
| (Activation)          | 192               |         |              |
|                       |                   |         |              |
| activation_74         | (None, 17, 17,    | 0       |              |
| batch_normalizat...   |                   |         |              |
| (Activation)          | 192               |         |              |
|                       |                   |         |              |
| conv2d_71 (Conv2D)    | (None, 8, 8, 320) | 552,960 |              |
| activation_70[0]...   |                   |         |              |
|                       |                   |         |              |
| conv2d_75 (Conv2D)    | (None, 8, 8, 192) | 331,776 |              |
| activation_74[0]...   |                   |         |              |
|                       |                   |         |              |
| batch_normalizatio... | (None, 8, 8, 320) | 960     |              |
| conv2d_71[0][0]       |                   |         |              |
| (BatchNormalizatio... |                   |         |              |
|                       |                   |         |              |
| batch_normalizatio... | (None, 8, 8, 192) | 576     |              |
| conv2d_75[0][0]       |                   |         |              |
| (BatchNormalizatio... |                   |         |              |
|                       |                   |         |              |
| activation_71         | (None, 8, 8, 320) | 0       |              |
| batch_normalizat...   |                   |         |              |
| (Activation)          |                   |         |              |
|                       |                   |         |              |
| activation_75         | (None, 8, 8, 192) | 0       |              |
| batch_normalizat...   |                   |         |              |
| (Activation)          |                   |         |              |
|                       |                   |         |              |
| max_pooling2d_3       | (None, 8, 8, 768) | 0       | mixed7[0][0] |
| (MaxPooling2D)        |                   |         |              |
|                       |                   |         |              |

|                       |                   |                         |
|-----------------------|-------------------|-------------------------|
| mixed8                | (None, 8, 8,      | 0                       |
| activation_71[0]...   | 1280)             |                         |
| (Concatenate)         |                   |                         |
| activation_75[0]...   |                   |                         |
|                       |                   |                         |
| max_pooling2d_3[...]  |                   |                         |
|                       |                   |                         |
| conv2d_80 (Conv2D)    | (None, 8, 8, 448) | 573,440   mixed8[0] [0] |
|                       |                   |                         |
| batch_normalizatio... | (None, 8, 8, 448) | 1,344                   |
| conv2d_80[0][0]       |                   |                         |
| (BatchNormalizatio... |                   |                         |
|                       |                   |                         |
| activation_80         | (None, 8, 8, 448) | 0                       |
| batch_normalizat...   |                   |                         |
| (Activation)          |                   |                         |
|                       |                   |                         |
| conv2d_77 (Conv2D)    | (None, 8, 8, 384) | 491,520   mixed8[0] [0] |
|                       |                   |                         |
| conv2d_81 (Conv2D)    | (None, 8, 8, 384) | 1,548,288               |
| activation_80[0]...   |                   |                         |
|                       |                   |                         |
| batch_normalizatio... | (None, 8, 8, 384) | 1,152                   |
| conv2d_77[0][0]       |                   |                         |
| (BatchNormalizatio... |                   |                         |
|                       |                   |                         |
| batch_normalizatio... | (None, 8, 8, 384) | 1,152                   |
| conv2d_81[0][0]       |                   |                         |
| (BatchNormalizatio... |                   |                         |
|                       |                   |                         |
| activation_77         | (None, 8, 8, 384) | 0                       |
| batch_normalizat...   |                   |                         |
| (Activation)          |                   |                         |
|                       |                   |                         |

|                       |                   |         |              |
|-----------------------|-------------------|---------|--------------|
| activation_81         | (None, 8, 8, 384) | 0       |              |
| batch_normalizat...   |                   |         |              |
| (Activation)          |                   |         |              |
|                       |                   |         |              |
| conv2d_78 (Conv2D)    | (None, 8, 8, 384) | 442,368 |              |
| activation_77[0]...   |                   |         |              |
|                       |                   |         |              |
| conv2d_79 (Conv2D)    | (None, 8, 8, 384) | 442,368 |              |
| activation_77[0]...   |                   |         |              |
|                       |                   |         |              |
| conv2d_82 (Conv2D)    | (None, 8, 8, 384) | 442,368 |              |
| activation_81[0]...   |                   |         |              |
|                       |                   |         |              |
| conv2d_83 (Conv2D)    | (None, 8, 8, 384) | 442,368 |              |
| activation_81[0]...   |                   |         |              |
|                       |                   |         |              |
| average_pooling2d_7   | (None, 8, 8,      | 0       | mixed8[0][0] |
| (AveragePooling2D)    | 1280)             |         |              |
|                       |                   |         |              |
| conv2d_76 (Conv2D)    | (None, 8, 8, 320) | 409,600 | mixed8[0][0] |
|                       |                   |         |              |
| batch_normalizatio... | (None, 8, 8, 384) | 1,152   |              |
| conv2d_78[0][0]       |                   |         |              |
| (BatchNormalizatio... |                   |         |              |
|                       |                   |         |              |
| batch_normalizatio... | (None, 8, 8, 384) | 1,152   |              |
| conv2d_79[0][0]       |                   |         |              |
| (BatchNormalizatio... |                   |         |              |
|                       |                   |         |              |
| batch_normalizatio... | (None, 8, 8, 384) | 1,152   |              |
| conv2d_82[0][0]       |                   |         |              |
| (BatchNormalizatio... |                   |         |              |
|                       |                   |         |              |

|                       |                       |         |
|-----------------------|-----------------------|---------|
| batch_normalizatio... | (None, 8, 8, 384)     | 1,152   |
| conv2d_83[0][0]       | (BatchNormalizatio... |         |
|                       |                       |         |
|                       |                       |         |
| conv2d_84 (Conv2D)    | (None, 8, 8, 192)     | 245,760 |
| average_pooling2...   |                       |         |
|                       |                       |         |
|                       |                       |         |
| batch_normalizatio... | (None, 8, 8, 320)     | 960     |
| conv2d_76[0][0]       | (BatchNormalizatio... |         |
|                       |                       |         |
|                       |                       |         |
|                       |                       |         |
| activation_78         | (None, 8, 8, 384)     | 0       |
| batch_normalizat...   | (Activation)          |         |
|                       |                       |         |
|                       |                       |         |
|                       |                       |         |
| activation_79         | (None, 8, 8, 384)     | 0       |
| batch_normalizat...   | (Activation)          |         |
|                       |                       |         |
|                       |                       |         |
|                       |                       |         |
| activation_82         | (None, 8, 8, 384)     | 0       |
| batch_normalizat...   | (Activation)          |         |
|                       |                       |         |
|                       |                       |         |
|                       |                       |         |
| activation_83         | (None, 8, 8, 384)     | 0       |
| batch_normalizat...   | (Activation)          |         |
|                       |                       |         |
|                       |                       |         |
|                       |                       |         |
| batch_normalizatio... | (None, 8, 8, 192)     | 576     |
| conv2d_84[0][0]       | (BatchNormalizatio... |         |
|                       |                       |         |
|                       |                       |         |
|                       |                       |         |
| activation_76         | (None, 8, 8, 320)     | 0       |
| batch_normalizat...   | (Activation)          |         |
|                       |                       |         |
|                       |                       |         |
|                       |                       |         |

|                       |                   |         |              |  |
|-----------------------|-------------------|---------|--------------|--|
|                       |                   |         |              |  |
| mixed9_0              | (None, 8, 8, 768) | 0       |              |  |
| activation_78[0]...   |                   |         |              |  |
| (Concatenate)         |                   |         |              |  |
| activation_79[0]...   |                   |         |              |  |
|                       |                   |         |              |  |
| concatenate           | (None, 8, 8, 768) | 0       |              |  |
| activation_82[0]...   |                   |         |              |  |
| (Concatenate)         |                   |         |              |  |
| activation_83[0]...   |                   |         |              |  |
|                       |                   |         |              |  |
| activation_84         | (None, 8, 8, 192) | 0       |              |  |
| batch_normalizat...   |                   |         |              |  |
| (Activation)          |                   |         |              |  |
|                       |                   |         |              |  |
| mixed9                | (None, 8, 8,      | 0       |              |  |
| activation_76[0]...   |                   |         |              |  |
| (Concatenate)         | 2048)             |         |              |  |
| mixed9_0[0][0],       |                   |         |              |  |
|                       |                   |         |              |  |
| concatenate[0][0...   |                   |         |              |  |
|                       |                   |         |              |  |
| activation_84[0]...   |                   |         |              |  |
|                       |                   |         |              |  |
| conv2d_89 (Conv2D)    | (None, 8, 8, 448) | 917,504 | mixed9[0][0] |  |
|                       |                   |         |              |  |
| batch_normalizatio... | (None, 8, 8, 448) | 1,344   |              |  |
| conv2d_89[0][0]       |                   |         |              |  |
| (BatchNormalizatio... |                   |         |              |  |
|                       |                   |         |              |  |
| activation_89         | (None, 8, 8, 448) | 0       |              |  |
| batch_normalizat...   |                   |         |              |  |
| (Activation)          |                   |         |              |  |
|                       |                   |         |              |  |
| conv2d_86 (Conv2D)    | (None, 8, 8, 384) | 786,432 | mixed9[0][0] |  |
|                       |                   |         |              |  |

|                       |                   |           |              |
|-----------------------|-------------------|-----------|--------------|
| conv2d_90 (Conv2D)    | (None, 8, 8, 384) | 1,548,288 |              |
| activation_89[0]...   |                   |           |              |
| batch_normalizatio... | (None, 8, 8, 384) | 1,152     |              |
| conv2d_86[0][0]       |                   |           |              |
| (BatchNormalizatio... |                   |           |              |
| batch_normalizatio... | (None, 8, 8, 384) | 1,152     |              |
| conv2d_90[0][0]       |                   |           |              |
| (BatchNormalizatio... |                   |           |              |
| activation_86         | (None, 8, 8, 384) | 0         |              |
| batch_normalizat...   |                   |           |              |
| (Activation)          |                   |           |              |
| activation_90         | (None, 8, 8, 384) | 0         |              |
| batch_normalizat...   |                   |           |              |
| (Activation)          |                   |           |              |
| conv2d_87 (Conv2D)    | (None, 8, 8, 384) | 442,368   |              |
| activation_86[0]...   |                   |           |              |
| conv2d_88 (Conv2D)    | (None, 8, 8, 384) | 442,368   |              |
| activation_86[0]...   |                   |           |              |
| conv2d_91 (Conv2D)    | (None, 8, 8, 384) | 442,368   |              |
| activation_90[0]...   |                   |           |              |
| conv2d_92 (Conv2D)    | (None, 8, 8, 384) | 442,368   |              |
| activation_90[0]...   |                   |           |              |
| average_pooling2d_8   | (None, 8, 8,      | 0         | mixed9[0][0] |
| (AveragePooling2D)    | 2048)             |           |              |

|  |                   |         |              |
|--|-------------------|---------|--------------|
| conv2d_85 (Conv2D)                             | (None, 8, 8, 320) | 655,360 | mixed9[0][0] |
| batch_normalizatio...                          | (None, 8, 8, 384) | 1,152   |              |
| conv2d_87[0][0]<br>(BatchNormalizatio...)      |                   |         |              |
| batch_normalizatio...                          | (None, 8, 8, 384) | 1,152   |              |
| conv2d_88[0][0]<br>(BatchNormalizatio...)      |                   |         |              |
| batch_normalizatio...                          | (None, 8, 8, 384) | 1,152   |              |
| conv2d_91[0][0]<br>(BatchNormalizatio...)      |                   |         |              |
| batch_normalizatio...                          | (None, 8, 8, 384) | 1,152   |              |
| conv2d_92[0][0]<br>(BatchNormalizatio...)      |                   |         |              |
| conv2d_93 (Conv2D)                             | (None, 8, 8, 192) | 393,216 |              |
| average_pooling2...                            |                   |         |              |
| batch_normalizatio...                          | (None, 8, 8, 320) | 960     |              |
| conv2d_85[0][0]<br>(BatchNormalizatio...)      |                   |         |              |
| activation_87<br>batch_norm...<br>(Activation) | (None, 8, 8, 384) | 0       |              |
| activation_88<br>batch_norm...<br>(Activation) | (None, 8, 8, 384) | 0       |              |

|                       |                   |     |
|-----------------------|-------------------|-----|
| activation_91         | (None, 8, 8, 384) | 0   |
| batch_normalizat...   |                   |     |
| (Activation)          |                   |     |
|                       |                   |     |
| activation_92         | (None, 8, 8, 384) | 0   |
| batch_normalizat...   |                   |     |
| (Activation)          |                   |     |
|                       |                   |     |
| batch_normalizatio... | (None, 8, 8, 192) | 576 |
| conv2d_93[0][0]       |                   |     |
| (BatchNormalizatio... |                   |     |
|                       |                   |     |
| activation_85         | (None, 8, 8, 320) | 0   |
| batch_normalizat...   |                   |     |
| (Activation)          |                   |     |
|                       |                   |     |
| mixed9_1              | (None, 8, 8, 768) | 0   |
| activation_87[0]...   |                   |     |
| (Concatenate)         |                   |     |
| activation_88[0]...   |                   |     |
|                       |                   |     |
| concatenate_1         | (None, 8, 8, 768) | 0   |
| activation_91[0]...   |                   |     |
| (Concatenate)         |                   |     |
| activation_92[0]...   |                   |     |
|                       |                   |     |
| activation_93         | (None, 8, 8, 192) | 0   |
| batch_normalizat...   |                   |     |
| (Activation)          |                   |     |
|                       |                   |     |
| mixed10               | (None, 8, 8,      | 0   |
| activation_85[0]...   |                   |     |
| (Concatenate)         | 2048)             |     |
| mixed9_1[0][0],       |                   |     |
|                       |                   |     |
| concatenate_1[0]...   |                   |     |
|                       |                   |     |
| activation_93[0]...   |                   |     |

|                       |              |           |               |
|-----------------------|--------------|-----------|---------------|
|                       |              |           |               |
| global_average_poo... | (None, 2048) | 0         | mixed10[0][0] |
| (GlobalAveragePool... |              |           |               |
|                       |              |           |               |
| dense (Dense)         | (None, 1024) | 2,098,176 |               |
| global_average_p...   |              |           |               |
|                       |              |           |               |
| dropout (Dropout)     | (None, 1024) | 0         | dense[0][0]   |
|                       |              |           |               |
| dense_1 (Dense)       | (None, 8)    | 8,200     | dropout[0][0] |
|                       |              |           |               |

**Total params:** 23,909,160 (91.21 MB)

**Trainable params:** 2,106,376 (8.04 MB)

**Non-trainable params:** 21,802,784 (83.17 MB)

-----  
Starting model training for 15 epochs...

```
d:\MHSSCE\Sem5\early_detection_of_dr\proj\.venv\Lib\site-packages\keras
\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning:
Your `PyDataset` class should call `super().__init__(**kwargs)` in its
constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
```

```
    self._warn_if_super_not_called()
```

Epoch 1/15  
**159/159** ————— **645s** 4s/step - accuracy:  
0.4641 - loss: 1.4874 - val\_accuracy: 0.5144 - val\_loss: 1.3071

Epoch 2/15

**1/159** ————— **6:21** 2s/step - accuracy:  
0.3125 - loss: 1.5021

```
d:\MHSSCE\Sem5\early_detection_of_dr\proj\.venv\Lib\site-packages\keras
\src\trainers\epoch_iterator.py:116: UserWarning: Your input ran out of
data; interrupting training. Make sure that your dataset or generator
```

```
can generate at least `steps_per_epoch * epochs` batches. You may need  
to use the `repeat()` function when building your dataset.
```

```
    self._interrupted_warning()
```

```
159/159 ━━━━━━━━━━ 101s 626ms/step - accuracy:  
0.3125 - loss: 1.5021 - val_accuracy: 0.5168 - val_loss: 1.3071  
Epoch 3/15  
159/159 ━━━━━━━━━━ 664s 4s/step - accuracy:  
0.5153 - loss: 1.2939 - val_accuracy: 0.5192 - val_loss: 1.2695  
Epoch 4/15  
159/159 ━━━━━━━━━━ 120s 738ms/step - accuracy:  
0.5312 - loss: 1.2767 - val_accuracy: 0.5112 - val_loss: 1.2696  
Epoch 5/15  
159/159 ━━━━━━━━━━ 536s 3s/step - accuracy:  
0.5219 - loss: 1.2726 - val_accuracy: 0.5160 - val_loss: 1.2623  
Epoch 6/15  
159/159 ━━━━━━━━━━ 92s 568ms/step - accuracy:  
0.6875 - loss: 1.1165 - val_accuracy: 0.5168 - val_loss: 1.2613  
Epoch 7/15  
159/159 ━━━━━━━━━━ 534s 3s/step - accuracy:  
0.5269 - loss: 1.2564 - val_accuracy: 0.5304 - val_loss: 1.3005  
Epoch 8/15  
159/159 ━━━━━━━━━━ 91s 562ms/step - accuracy:  
0.4688 - loss: 1.3154 - val_accuracy: 0.5280 - val_loss: 1.2870  
Epoch 9/15  
159/159 ━━━━━━━━━━ 517s 3s/step - accuracy:  
0.5292 - loss: 1.2528 - val_accuracy: 0.5288 - val_loss: 1.2629  
Epoch 10/15  
159/159 ━━━━━━━━━━ 89s 550ms/step - accuracy:  
0.7188 - loss: 0.9228 - val_accuracy: 0.5256 - val_loss: 1.2742  
Epoch 11/15  
159/159 ━━━━━━━━━━ 532s 3s/step - accuracy:  
0.5300 - loss: 1.2470 - val_accuracy: 0.5240 - val_loss: 1.2626  
Epoch 12/15  
159/159 ━━━━━━━━━━ 92s 566ms/step - accuracy:  
0.5312 - loss: 1.1893 - val_accuracy: 0.5184 - val_loss: 1.2671  
Epoch 13/15  
159/159 ━━━━━━━━━━ 566s 4s/step - accuracy:  
0.5330 - loss: 1.2394 - val_accuracy: 0.5232 - val_loss: 1.2578  
Epoch 14/15  
159/159 ━━━━━━━━━━ 92s 570ms/step - accuracy:  
0.4375 - loss: 1.2752 - val_accuracy: 0.5312 - val_loss: 1.2432  
Epoch 15/15  
159/159 ━━━━━━━━━━ 580s 4s/step - accuracy:  
0.5341 - loss: 1.2260 - val_accuracy: 0.5329 - val_loss: 1.2571  
Training complete.  
-----  
Evaluating model on the validation set...  
40/40 ━━━━━━━━━━ 105s 3s/step - accuracy: 0.5340  
- loss: 1.2576
```

```
Validation Loss: 1.2576
Validation Accuracy: 0.5340
-----
Model successfully saved to: fundus_inceptionv3_model.keras
```

The Kernel crashed while executing code in the current cell or a previous cell.

Please review the code in the cell(s) to identify a possible cause of the failure.

Click [here](https://aka.ms/vscodeJupyterKernelCrash) for more info.

View Jupyter [log](command:jupyter.viewOutput) for further details.