**Report**

**ECMM426 - Computer Vision**

**Name：Zhenwei Song**

This report outlines the output filenames and results for each question in the assignment.

**Question 1:**

Required Output: Two 2-dimensional numpy arrays of the same shape as the input image.

Output Files:

gradient\_magnitude.npy: Numpy array containing gradient magnitude values

gradient\_direction.npy: Numpy array containing gradient direction values

Additional Files:

gradient\_magnitude.jpg: Visualization of gradient magnitude

gradient\_direction.jpg: Visualization of gradient direction

gradient\_visualization.jpg: Combined visualization of original image, gradient magnitude, and gradient direction

Array Shapes:

Both arrays have shape (606, 768) matching the input image dimensions

**Question 2:**

Required Output: Three 1-dimensional numpy arrays for R, G, and B channels.

Output Files:

histogram\_r.npy: Numpy array containing histogram for red channel

histogram\_g.npy: Numpy array containing histogram for green channel

histogram\_b.npy: Numpy array containing histogram for blue channel

Additional Files:

color\_histograms.jpg: Visualization of the color histograms for all three channels

Array Properties:

Each histogram has shape (32,) representing 32 bins

Red histogram: Contains pixel counts in low-mid value range, with peaks around bins 1-2

Green histogram: Contains pixel counts across wide range, with bimodal distribution

Blue histogram: Contains pixel counts with peaks around bins 2-3

**Question 3:**

Required Output: A single 3x3 transformation matrix.

Output Files:

transform\_matrix.npy: Numpy array containing the 3x3 transformation matrix

Additional Files:

transformation\_visualization.jpg: Visualization of original and transformed points

Matrix Values:

[[ 1.06066017 -1.06066017 732.02778552]

[ 1.06066017 1.06066017 -550.01466213]

[ 0. 0. 1. ]]

**Question 4:**

Required Output: Trained model weights.

Output Files:

data/weights\_resnet.pth: PyTorch model weights file

Training Results:

Best validation accuracy: 79.53%

Final training accuracy: 82.38%

**Question 5:**

Required Outputs:

A 1-dimensional numpy array of predicted labels

Classification accuracy

Output Files:

test\_predictions.npy: Numpy array containing predicted class labels

confusion\_matrix.png: Visualization of the confusion matrix

Results:

Test accuracy: 79.75%

Number of predictions: 6000 (all animal classes in test set)

**Question 6:**

Required Outputs:

A 2-dimensional numpy array of mask counts

MAPE score

Trained model weights

Output Files:

data/weights\_counting.pth: PyTorch model weights file

Results:

MAPE score: 15.89%

Counts array: 2D array of shape (n, 3) containing counts of with\_mask, without\_mask, and mask\_weared\_incorrect classes for each image