Image Processing using Fourier Transforms in Cancer Image Classification: Fourier ViT Model

Samridhi Raj Sinha
B074 Btech CE

Research Paper

Fourier ViT: A Multi-scale Vision Transformer with Fourier Transform for Histopathological Image Classification

1 Summary

The research paper introduces a novel approach that combines Fourier transform with Vision Transformer (ViT) to enhance the classification of unstructured histopathological images. The study addresses the challenge of accurately classifying medical images, particularly histopathological images that are highly detailed and unstructured compared to natural images. By integrating Fourier transform to extract fine-grained details and ViT for strong attention mechanisms, the proposed Fourier ViT model demonstrates superior performance over classic CNN-based and Transformer-based models in histopathological image classification.

Future scope for Fourier transform in medical research includes its potential application in other medical imaging tasks beyond classification, such as image segmentation and detection. The integration of multi-domain features, as demonstrated in Fourier ViT, can enhance the efficiency of models in learning complex medical image features. Moreover, the study suggests that Fourier transform can be utilized as a preprocessing step in different datasets, like melanoma datasets, to improve feature extraction and classification accuracy. By leveraging the complementary strengths of Fourier transform and deep learning models like ViT, advancements in medical image analysis and diagnosis can be achieved with greater precision and efficiency.

2 Problem: Solving 2D Discrete Fourier Transform (DFT)

Given an $N \times N$ matrix A, the 2D Discrete Fourier Transform (DFT) of A, denoted as F, is calculated as follows:

Data: A: input matrix of size $N \times N$
Result: F: 2D DFT of A
for $u = 0$ **to** $N - 1$ **do**
  for $v = 0$ **to** $N - 1$ **do**
    $F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} A(x, y) e^{-2\pi i(\frac{ux}{N} + \frac{vy}{N})}$;
  end
end

Algorithm 1: 2D Discrete Fourier Transform

Here, $F(u, v)$ represents the transformed coefficient at position (u, v) in the frequency domain, and $A(x, y)$ represents the pixel value at position (x, y) in the spatial domain.

3 Application

We have implemented the Fourier transform as a preprocessing tool for the Fourier ViT model on a melanoma dataset that contains 5000 benign and malignant images each. Our end goal is to create a more accurate model than can classify benign and malignant images.

The Python code below demonstrates how we applied the Fourier transform as a preprocessing step to manipulate the Y channel of the YCbCr color space in the images:

```python
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt

image_dir = r"/kaggle/input/melanoma-skin-cancer-dataset-of-10000-images/melanoma_cancer_dataset/train/malignant"

image_files = os.listdir(image_dir)

# Iteration
for i, image_file in enumerate(image_files):
    # Load the image
    image_path = os.path.join(image_dir, image_file)
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # fourier
    magnitude_spectrum = np.abs(np.fft.fftshift(np.fft.fft2(image)))
    phase_spectrum = np.angle(np.fft.fftshift(np.fft.fft2(image)))

    # log scaling
    magnitude_spectrum_log = np.log1p(magnitude_spectrum)
```

```python

    # Plotting
    plt.figure(figsize=(20, 5))

    plt.subplot(1, 4, 1)
    plt.imshow(image, cmap='gray')
    plt.title("Original Image")
    plt.axis('off')

    plt.subplot(1, 4, 2)
    plt.imshow(magnitude_spectrum_log, cmap='gray')
    plt.title("Magnitude Spectrum (Log Scale)")
    plt.axis('off')
    plt.xticks([])
    plt.yticks([])

    plt.subplot(1, 4, 3)
    plt.imshow(phase_spectrum, cmap='gray')
    plt.title("Phase Spectrum")
    plt.axis('off')
    plt.xticks([])
    plt.yticks([])


    colored_image = cv2.imread(image_path)
    colored_image_rgb = cv2.cvtColor(colored_image, cv2.
    COLOR_BGR2RGB)

    plt.subplot(1, 4, 4)
    plt.imshow(colored_image_rgb)
    plt.title("Colored Image")
    plt.axis('off')

    plt.show()
```

Listing 1: Python code for Fourier transformation

This code applies a horizontal mask to the Fourier-transformed Y channel of the YCbCr color space for a given input image. The masked Fourier transform is then inverted to obtain the modified Y channel, which is combined with the original Cr and Cb channels to reconstruct the transformed color image. This approach demonstrates how Fourier transforms can be used for preprocessing medical images, such as those in the melanoma dataset, to enhance feature extraction and improve classification accuracy in cancer image classification tasks.

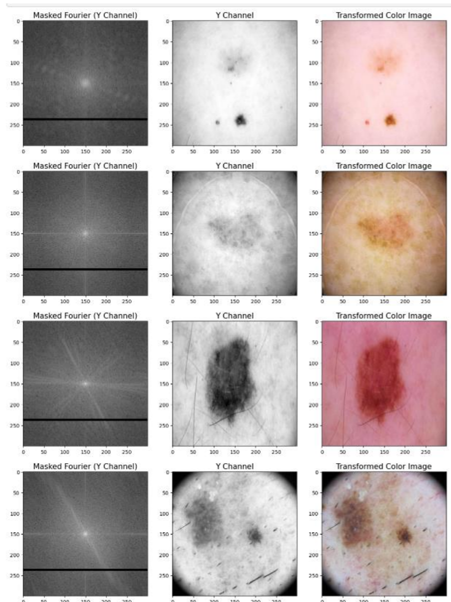

Figure 1: Output image after applying Fourier transformation

References

1. **Fourier transform python notebook** `https://www.kaggle.com/code/samrs22/image-classification-fourier-transform/edit/run/164273364`

2. **Melanoma Skin Cancer Dataset of 10000 Images**
   Hasnain Javed
   `https://www.kaggle.com/datasets//melanoma-skin-cancer-dataset-of-10000-images`

3. **Research paper citation**
   Authors
   `Fourier_ViT_A_Multi-scale_Vision_Transformer_with__240313_231314.`