AMRITA
VISHWA VIDYAPEETHAM

Art Style Classification

Team members:

Aparna S.	CB.AI.U4AIM24104
Rubashree R.	CB.AI.U4AIM24139
Sakthikailash SV.	CB.AI.U4AIM24140
Sangeetha S.	CB.AI.U4AIM24142

Introduction

This project aims to classify artworks into different styles using a hybrid approach that combines Fourier Transform for feature extraction and CNNs for classification. By leveraging frequency-domain analysis, we enhance computational efficiency while maintaining high accuracy in art classification.

Problem Statement

Art classification is challenging due to diverse styles and the high computational cost of traditional CNNs. This project proposes a hybrid FFT-CNN approach to reduce dimensionality and improve efficiency, enabling faster, scalable, and accurate classification of large art datasets. The goal is real-time performance.

Dataset

- Link:

  https://www.kaggle.com/datasets/cyanex1702/surreal-symphonies-a-dataset-of-diverse-art
- Total Images: 7,320
- Number of Classes: 30 (Each class represents a different art style)
- File Format: .jpg

LITERATURE SURVEY

S.NO	TITLE	AUTHORS	METHODOLOGY
1	Fast Fourier Transforms – A Tutorial Review and State of the Art	P. Duhamel, M. Vetterli	The Fast Fourier Transform (FFT) accelerates DFT computation, with applications in signal processing, machine learning, and modern computing advancements.
2	A quantitative approach to painting styles	Vilson Vieira, Renato Fabbri, David Sbrissa, Luciano da Fontoura Costa, Gonzalo Travieso	The paper uses image processing, statistical analysis, and machine learning to quantify and classify painting styles.
3	Artistic Style Recognition: Combining Deep and Shallow Neural Networks for Painting Classification	Saqib Imran , Rizwan Ali Naqvi , Muhammad Sajid , Tauqeer Safdar Malik , Saif Ullah , Syed Atif Moqurrab , and D	The paper combines deep and shallow neural networks to enhance painting classification by effectively extracting features and improving recognition of artistic styles.
4	Recognizing Art Style Automatically in Painting Using Convolutional Neural Network	Mahfuza Akter, Mst. Rasheda Akther & Md. Khaliluzzaman	The paper utilizes Convolutional Neural Networks (CNNs) for automatic recognition and classification of art styles in paintings.

Methodology

1. Dataset Collection.
2. Data Augmentation.
3. Train-validation-Test Split.
4. Fourier Transform (FFT).
5. CNN Model Training.
6. Model Evaluation & Optimization.
7. Model Deployment.

Data Structures

- List:

```
images = []
for root, _, files in os.walk(class_source_path):
    images.extend([os.path.join(root, img) for img in files if img.lower().endswith(('jpg', 'png', 'jpeg'))])
```

- Numpyarray:

```
image = cv2.imread(image_path)
```

- Dictionary:

```
augmented = transform(image=image)['image']
```

- Tuple:

```
img_size=(224, 224)
```

```
lab = cv2.cvtColor(augmented, cv2.COLOR_RGB2LAB)
l, a, b = cv2.split(lab)
```

- String:

```
input_folder = r"C:\Users\SANGEETHA S\Downloads\maths dataset\Surreal Symphonies (A dataset of Diverse Artistic Visions)"
output_folder = r"C:\Users\SANGEETHA S\Downloads\maths Augmentation images"
```

Algorithms

- Probabilistic Data Augmentation Algorithm

1. Input: Original Image
2. Process: Apply Transformations in Sequence:
   a. Rotation
   b. Horizontal Flip
   c. Vertical Flip
   d. Color Jittering
   e. Elastic Transformation
3. Output: Augmented Image

```
transform = A.Compose([
    A.Rotate(limit=30, p=0.7),
    A.HorizontalFlip(p=0.5),
    A.VerticalFlip(p=0.3),
    A.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1, p=0.7),
    A.ElasticTransform(alpha=1, sigma=50, p=0.5),
])
```

- **Contrast Limited Adaptive Histogram Equalization (CLAHE)**

1. Convert the input image from RGB to LAB color space.
2. Split the LAB image into three channels: L (Luminance), A, and B.
3. Create a CLAHE object with:
   - clipLimit = 2.0 (limits contrast amplification to reduce noise)
   - tileGridSize = (8,8) (divides the image into smaller regions for adaptive equalization)
4. Apply CLAHE only to the Luminance (L) channel to enhance brightness.
5. Merge the modified Luminance (L_eq) channel back with the original A and B channels.
6. Convert the image from LAB back to RGB color space.
7. Return the final contrast-enhanced image.

```
lab = cv2.cvtColor(augmented, cv2.COLOR_RGB2LAB)
l, a, b = cv2.split(lab)
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
l_eq = clahe.apply(l)
lab_eq = cv2.merge((l_eq, a, b))
final_image = cv2.cvtColor(lab_eq, cv2.COLOR_LAB2RGB)
```

Fast Fourier Transform (FFT) Algorithm.

1. Input: Grayscale Image

2. Apply Fourier Transform:
   a. Compute the 2D Fast Fourier Transform (FFT) using fft2().
   b. Shift the zero-frequency component to the center using fftshift().

3. Compute Magnitude Spectrum:
   a. Take the absolute value of the shifted FFT result using np.abs().
   b. Apply logarithmic scaling using log(1 + magnitude) for better visualization.

4. Normalize the Magnitude Spectrum:
   a. Normalize values to the range 0-255 using cv2.normalize().
   b. Convert the result to uint8 format.

5. Output: Magnitude Spectrum Image.

```
f = np.fft.fft2(image)
fshift = np.fft.fftshift(f)
magnitude_spectrum = np.log(1 + np.abs(fshift))
```

Mathematical Concept:

DFT Formula: $$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

FFT speeds up DFT by breaking it into smaller parts using the divide-and-conquer method and also Reduces computation .

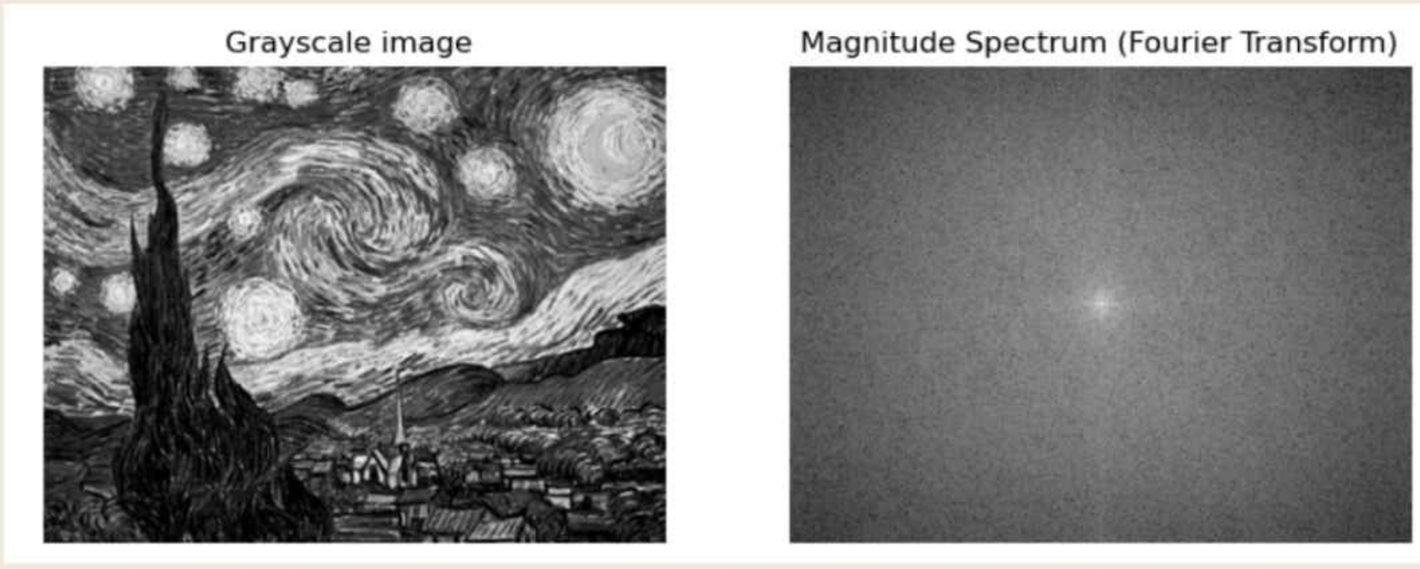
FFT Formula:

$$X[k] = E[k] + W_N^k O[k]$$

$$X[k + N/2] = E[k] - W_N^k O[k]$$

DFT	FFT
O(N^2)	O(N log N)
50 million operations	10,000 operations

WHAT WE HAVE DONE



Summary

This project uses Fourier Transform to extract frequency-based features from images and a CNN model for classification. The combination enhances pattern recognition and improves the accuracy of artwork classification.