# Industrial Project 2024 Part 02

Comprehensive Guide for Project Execution and Usage

**Group A:**

Sonain Jamil, Kasem Amnuayrotchanachinda, Muhammad Turab

sonainjamil@ieee.org

November 21, 2024

# Contents

# 1 Introduction

The **Industrial Project 2024 Part 02** focuses on processing spectral image data through image stitching and Principal Component Analysis (PCA). This project enables the assembly of multiple spectral images into a single, high-resolution stitched image and performs dimensionality reduction on the stitched data using PCA.

# 2 Project Features

- **Image Stitching:** Combines multiple hyperspectral images into a single, high-resolution stitched image.
- **Transformation Options:** Applies transformations to the stitched image before PCA, including:
  - No Transformation
  - Standard Normal Transformation
  - First Derivative (with contrast enhancement)
  - Second Derivative (with contrast enhancement)
- **PCA on Spectral Data:** Reduces the dimensionality of the stitched hyperspectral image, saving each principal component as an individual grayscale PNG image.
- **Configurable Parameters:** Utilizes a `config.yaml` file to set paths, transformation options, and other processing parameters.

# 3 Installation and Setup

## 3.1 Requirements

- Python 3.8 or higher.
- Required Python libraries:
  - `numpy`
  - `matplotlib`
  - `PyYAML`
  - `opencv-python`
  - `scikit-learn`

### 3.2 Setup Instructions

1. Clone the repository to your local machine:

```
git clone https://github.com/sonainjameel/
    Industrial_Project_2024_Part2.git
cd Industrial_Project_2024_Part2
```

2. Install the required dependencies:

```
pip install -r requirements.txt
```

3. Ensure you have the correct input data in the directories specified in `config.yaml`.

## 4 Configuration

The project relies on a `config.yaml` file for setting input paths and processing parameters. Below is an example configuration:

```yaml
paths:
  input_dir: "data/input/images" # Directory containing input
     hyperspectral images
  stitched_output: "data/stitched/stitched_image.hdr" # Path to save
     stitched hyperspectral image
  pca_output_dir: "data/pca_output" # Directory for saving PCA
     component images
parameters:
  stitching:
    num_columns: 8 # Number of images per row in the stitched output
  pca:
    n_components: 20 # Number of PCA components to keep
  transformation: "first_derivative" # Options: "none", "
     standard_normal", "first_derivative", "second_derivative"
  lower_percentile: 2 # Percentile for contrast enhancement in
     derivatives
  upper_percentile: 98
```

Listing 1: Example config.yaml File

## 5 Usage

To run each part of the pipeline, specify the configuration file and task (`stitch` or `pca`) in the command line.

### 5.1 Stitching

Run the stitching process to create a stitched hyperspectral image from the input directory:

```
python main.py config.yaml stitch
```

Listing 2: Run the Stitching Process

The stitched image will be saved at the location specified in the `stitched_output` path of the `config.yaml` file.

### 5.2 PCA Processing

Perform PCA on the stitched image to reduce dimensionality and generate principal component images:

```
python main.py config.yaml pca
```

Listing 3: Run the PCA Process

The PCA component images will be saved in the directory specified in the `pca_output_dir` field of the `config.yaml` file.

## 6 Project Structure

```
Industrial_Project_2024_Part2/
|-- config.yaml                    # Configuration file for paths and parameters
|-- main.py                        # Main script to run the pipeline
|-- requirements.txt               # List of dependencies
|-- data/                          # Directory for input and output data
|   |-- input/                     # Folder containing input hyperspectral images
|   |-- stitched/                  # Folder to save stitched output
|   |-- pca_output/                # Folder to save PCA component images
|-- modules/                       # Core project modules
    |-- __init__.py                # Initialize module
    |-- stitching.py               # Functions for image stitching
    |-- pca_processing.py          # Functions for PCA
    |-- transformations.py         # Functions for spectral transformations
```

## 7 Key Modules

- `stitching.py`: Handles the assembly of input spectral images into a single stitched image.

- `pca_processing.py`: Performs dimensionality reduction using PCA, saving each principal component as a grayscale image.
- `transformations.py`: Implements transformations such as standard normalization, first derivative, and second derivative with optional contrast enhancement.

## 8 Testing and Results

The following test cases were conducted:

- Verified stitching functionality with different layouts and resolutions.
- Tested PCA component generation with varying numbers of components.
- Ensured accuracy and clarity of transformed spectral data using different transformations.

## 9 Example Results

Below are some example outputs from the project:

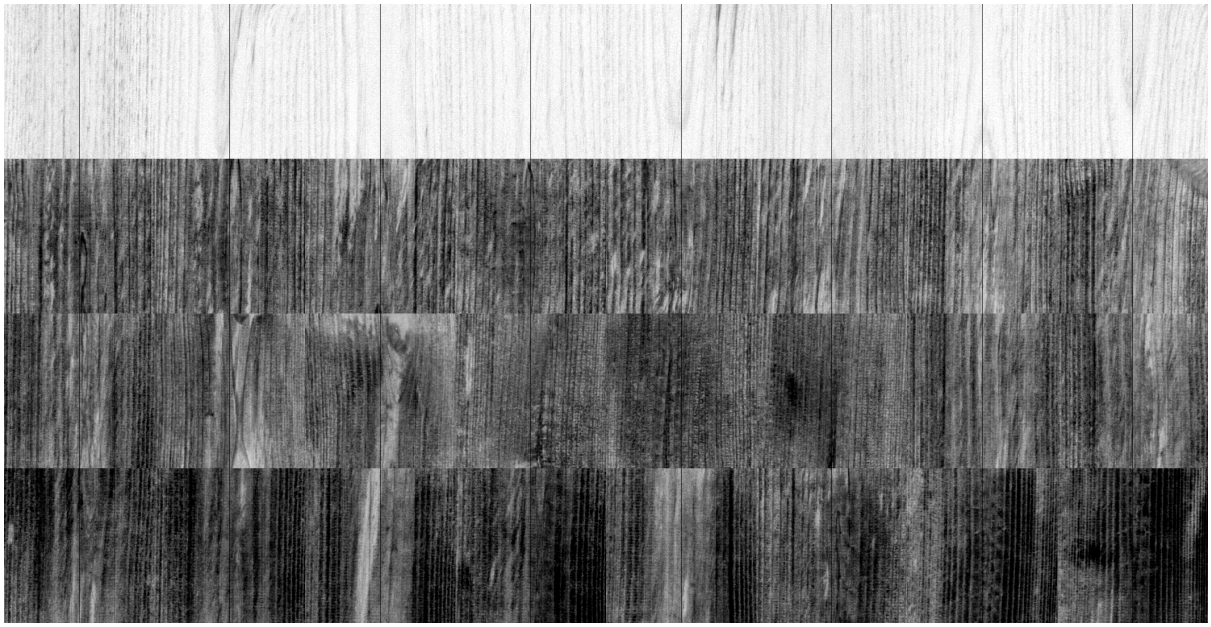

Figure 1: Principal Component 1 output from PCA.

## 10 References


- Python Documentation: https://www.python.org/doc/
- Libraries: NumPy, OpenCV, Matplotlib, PyYAML, Scikit-Learn.