

Single Responsibility Principle

Every class should focus on doing only one thing.

without SRP:

```
import os
import yaml
import cv2

class DataLoader:
    def __init__(self):
        BASE_PATH = os.path.dirname(__file__)
        with open(f"{BASE_PATH}/../config.yaml", "r") as file:
            cfg = yaml.safe_load(file)

        self.data_root = cfg['data']['root_dir']
        self.img_dir = os.path.join(self.data_root, "images")
        self.label_dir = os.path.join(self.data_root, "labels")

    def load_image(self, filename):
        image_path = os.path.join(self.img_dir, filename)
        img = cv2.imread(image_path)
        return img
```

with SRP

Splitting the responsibilities.

1. Only loads configuration

```
import os
import yaml

class ConfigLoader:
    def __init__(self, config_path=None):
        base_path = os.path.dirname(__file__)
        self.config_path = config_path or os.path.join(base_path,
            "../config.yaml")

    def load(self):
        with open(self.config_path, "r") as file:
            return yaml.safe_load(file)
```

2. Only manages folder paths

```
import os

class PathManager:
    def __init__(self, config):
        self.data_root = config['data']['root_dir']
        self.img_dir = os.path.join(self.data_root, "images")
        self.label_dir = os.path.join(self.data_root, "labels")
```

3. Only loads data

```
import os

class DataLoader:
    def __init__(self, img_dir, label_dir):
        self.img_dir = img_dir
        self.label_dir = label_dir

    def load_image(self, filename):
        path = os.path.join(self.img_dir, filename)
        return cv2.imread(path)
```

Now each class has one objective.