SVM Image Recognition: Female vs Male

โปรเจกต์นี้เป็น Mini Project สำหรับการเรียนรู้ **Machine Learning ด้วย Support Vector Machine (SVM)** โดยใช้การประมวลผลภาพ (Image Processing) เพื่อจำแนกรูปภาพออกเป็น 2 คลาส ได้แก่ **Female** และ **Male**

📌 Project Overview

โปรเจกต์ใช้ **SVM (Support Vector Machine)** ในการจำแนกภาพ โดยมีขั้นตอนการทำงานหลักดังนี้

```
Image Dataset
  ↓
Load Images
  ↓
Preprocessing
  ↓
Resize Images
  ↓
Flatten Image Features
  ↓
Train/Test Split
  ↓
SVM Training
  ↓
Prediction
  ↓
Evaluation
```

📂 Dataset Structure

Dataset แบ่งออกเป็น Training และ Validation โดยมีโครงสร้างดังนี้:

```
archive/
├── Training/
│   ├── female/
│   └── male/
|
└── Validation/
    ├── female/
    └── male/
```

สำหรับการฝึกโมเดล โปรแกรมจะใช้ข้อมูลจาก:

```
archive/Training/
```

โดยแบ่งออกเป็น 2 classes:

- `female`
- `male`

  หมายเหตุ: Dataset ไม่ได้รวมอยู่ใน Repository หากมีข้อจำกัดด้านขนาดหรือ License ของ Dataset

---

🛠️ Technologies Used

- **Python**
- **OpenCV (`cv2`)** — สำหรับอ่านและประมวลผลรูปภาพ
- **NumPy** — สำหรับจัดการข้อมูลตัวเลขและ Array
- **Scikit-learn** — สำหรับสร้างและฝึก SVM
- **Matplotlib** — สำหรับแสดงผลข้อมูลและกราฟ
- **Git / GitHub** — สำหรับจัดเก็บ Source Code

📁 Project Structure

```
classification/
│
├── main.py
├── data_loader.py
├── preprocessing.py
├── split_data.py
├── svm_model.py
│
├── outputs/
│   ├──── ...
│
└── README.md
```

⚙️ Installation

1. Clone Repository

```
git clone <YOUR-GITHUB-REPOSITORY-URL>
```

เข้าสู่โฟลเดอร์โปรเจกต์:

```
cd classification
```

2. Install Required Libraries

ติดตั้ง Library ที่จำเป็น:

```
pip install numpy opencv-python scikit-learn matplotlib
```

หรือถ้ามีไฟล์ `requirements.txt`:

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

▶️ How to Run

กำหนดตำแหน่ง Dataset ใน `main.py`:

```
DATA_PATH = r"C:\ML-CPE\LAB05\archive\Training"
```

จากนั้นรัน:

```
python main.py
```

โปรแกรมจะทำงานตามขั้นตอน:

```
[Step 1] Loading dataset...
[Step 2] preprocess images...
[Step 3] Splitting dataset...
[Step 4] Training SVM...
[Step 5] Evaluation...
```

🔄 Processing Pipeline

Step 1 — Loading Dataset

โปรแกรมจะอ่านรูปภาพจาก:

```
archive/Training/female
archive/Training/male
```

และกำหนด Label:

```
female → Class 0
male   → Class 1
```

Step 2 — Image Preprocessing

รูปภาพจะถูกนำมา Resize ให้มีขนาดเท่ากันก่อนนำไปใช้กับ Machine Learning

ตัวอย่างเช่น:

```
100 × 100 pixels
```

จากนั้นแปลงภาพให้เป็น Feature Vector

$$100 \times 100$$

```
  ↓
10,000 Features
```

ดังนั้น Feature Shape จะมีลักษณะ:

```
(number_of_images, 10000)
```

Step 3 — Train/Test Split

ข้อมูลจะถูกแบ่งออกเป็น:

```
Training Data
   +
Testing Data
```

โดยใช้ `train_test_split()` จาก Scikit-learn

และใช้:

```
stratify=y
```

เพื่อรักษาสัดส่วนของแต่ละ Class

Step 4 — SVM Training

โมเดลใช้:

```
Support Vector Machine (SVM)
```

เพื่อเรียนรู้ความแตกต่างระหว่าง:

```
Female vs Male
```

แนวคิดของ SVM คือการหา **Hyperplane** ที่สามารถแบ่งข้อมูลของแต่ละ Class ออกจากกันได้ดีที่สุด โดยพยายามเพิ่มระยะห่าง (Margin) ระหว่างกลุ่มข้อมูล

Step 5 — Prediction

หลังจาก Train โมเดลแล้ว จะนำข้อมูล Test ไปให้โมเดลทำนาย:

```
Input Image
  ↓
Preprocessing
  ↓
Feature Extraction
  ↓
SVM Model
  ↓
Prediction
  ↓
Female / Male
```

📊 Evaluation

สามารถประเมินประสิทธิภาพของโมเดลด้วย:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

ตัวอย่าง:

```
Accuracy: 0.XX
```

ค่า Accuracy ที่สูงหมายถึงโมเดลสามารถจำแนกภาพในชุดทดสอบได้ถูกต้องมากขึ้น

🧠 What is SVM?

Support Vector Machine (SVM) เป็น Machine Learning Algorithm สำหรับ Classification

หลักการสำคัญคือการหาเส้นแบ่งหรือ **Decision Boundary** ที่สามารถแบ่งข้อมูลออกเป็นกลุ่มต่าง ๆ โดยให้มีระยะห่างจากข้อมูลแต่ละกลุ่มมากที่สุด

สำหรับโปรเจกต์นี้:

```
            Decision Boundary
        |
     Female        |        Male
  ●●●    |    ○○○
 ●●●●    |   ○○○○
  ●●     |    ○○
        |
```

SVM จะเรียนรู้รูปแบบของ Feature จากภาพ เพื่อใช้ตัดสินว่าภาพใหม่อยู่ใน Class ใด

📌 Example Output

ตัวอย่างผลการทำงานของโปรแกรม:

```
SVM Image Recognition: Female vs Male

[Step 1] Loading dataset...
Detected classes: ['female', 'male']

Loaded class female: XXX images
Loaded class male: XXX images

Dataset loaded successfully.
Total images : XXX
Classes      : ['female', 'male']

[Step 2] preprocess images...
Feature shape: (XXX, 10000)

[Step 3] Splitting dataset...

[Step 4] Training SVM...

[Step 5] Evaluation...
```

```
Accuracy: XX.XX%
```

⚠️ Limitations

ประสิทธิภาพของโมเดลขึ้นอยู่กับ:

1. จำนวนรูปภาพใน Dataset
2. คุณภาพของรูปภาพ
3. ความหลากหลายของข้อมูล
4. ขนาดของภาพ
5. วิธีการ Preprocessing
6. SVM Parameters

หาก Dataset มีจำนวนน้อย อาจทำให้เกิด **Overfitting** และทำให้โมเดลทำงานได้ดีเฉพาะกับข้อมูลที่ใช้ในการฝึก

🚀 Future Improvements

สามารถพัฒนาโปรเจกต์เพิ่มเติมได้ เช่น:

- เพิ่มจำนวนรูปภาพใน Dataset
- ใช้ Data Augmentation
- ทดลองเปลี่ยน `C`
- ทดลองเปลี่ยน Kernel เช่น `linear`, `rbf`
- ใช้ HOG Feature แทนการ Flatten Pixel โดยตรง
- ทดลองใช้ PCA เพื่อลดจำนวน Features
- เพิ่ม Confusion Matrix
- เพิ่ม GUI สำหรับ Upload รูปภาพ
- เพิ่มระบบ Prediction รูปภาพใหม่

👨‍💻 Author

Machine Learning Course — SVM Mini Project

📄 License

โปรเจกต์นี้จัดทำขึ้นเพื่อการศึกษาและการเรียนรู้เกี่ยวกับ Machine Learning และ SVM