

Image Preprocessing Module Structure

To view this diagram in VSCode, you'll need:

1. Install the "PlantUML" extension by jebbs
2. Install Graphviz (required for PlantUML)
 - Windows: Download from [Graphviz website](#)
 - Mac: `brew install graphviz`
 - Linux: `sudo apt install graphviz`
3. Install "Markdown Preview Enhanced" extension

Class Diagram

```
' Use @startmindmap instead of @startuml for some VSCode extensions
@startuml

skinparam classAttributeIconSize 0

package "Image Preprocessing Module" {
    abstract class Config {
        + logger: Logger
        + input_directory: str
        + output_directory: str
        + embeddings_file: str
    }

    class ImagePreprocessor {
        - detector: FaceDetector
        - recognizer: FaceRecognizer
        + process_input_image(image_path: str): tuple
        + process_database_images(): tuple
    }

    class FaceDetector {
        + detect_faces(image: ndarray): list
    }

    class FaceRecognizer {
        + recognize_face(face_image: ndarray): ndarray
    }

    class Utils <<utility>> {
        + {static} contains_one_person(detections: list): bool
        + {static} extract_face(image: ndarray, detections: list): ndarray
        + {static} add_person_to_database(database: dict, person: str,
embeddings: list): None
    }

    Config <|-- ImagePreprocessor : inherits
```

```
ImagePreprocessor *-- FaceDetector : contains
ImagePreprocessor *-- FaceRecognizer : contains
ImagePreprocessor ..> Utils : uses
```

```
note right of ImagePreprocessor
    Main workflow:
    1. Load image from path
    2. Validate dimensions (≥640x640)
    3. Detect faces using detector
    4. Validate single face present
    5. Extract face region
    6. Generate face embeddings
    7. Save processed face & embeddings
end note
```

```
}
```

```
@enduml
```

Alternative Approach Using Mermaid

If you can't get PlantUML working, you can use Mermaid which is supported natively by GitHub and many Markdown editors:

```
classDiagram
    class Config {
        +logger: Logger
        +input_directory: str
        +output_directory: str
        +embeddings_file: str
    }

    class ImagePreprocessor {
        -detector: FaceDetector
        -recognizer: FaceRecognizer
        +process_input_image(image_path: str) tuple
        +process_database_images() tuple
    }

    class FaceDetector {
        +detect_faces(image: ndarray) list
    }

    class FaceRecognizer {
        +recognize_face(face_image: ndarray) ndarray
    }

    class Utils {
        +contains_one_person(detections: list) bool
        +extract_face(image: ndarray, detections: list) ndarray
        +add_person_to_database(database: dict, person: str, embeddings:
list) void
```

```
}

Config <|-- ImagePreprocessor : inherits
ImagePreprocessor *-- FaceDetector : contains
ImagePreprocessor *-- FaceRecognizer : contains
ImagePreprocessor ..> Utils : uses
```

Installation Troubleshooting

If you're still having trouble rendering the PlantUML diagram:

1. Verify PlantUML extension settings:

```
{
  "plantuml.render": "PlantUMLServer",
  "plantuml.server": "https://www.plantuml.com/plantuml",
}
```

2. Try using the online PlantUML editor:

- Copy the diagram code between `@startuml` and `@enduml`
- Paste it at [PlantUML Web Server](https://www.plantuml.com/plantuml)

3. Check if Java is installed (required by some PlantUML extensions):

```
java --version
```

4. For local rendering, ensure your PATH includes Graphviz:

- Windows: Add Graphviz's `bin` directory to PATH
- Unix: `echo $PATH | grep "graphviz"`