

Image Preprocessing Module Structure

PlantUML Class Diagram

```
@startuml

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

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

    ' Face detection component
    class FaceDetector {
        + detect_faces(image: ndarray): list
    }

    ' Face recognition component
    class FaceRecognizer {
        + recognize_face(face_image: ndarray): ndarray
    }

    ' Utility functions
    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
    }

    ' Define relationships
    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
```

PlantUML Syntax Explanation

Class Definitions

```
class ClassName {
    + public_attribute
    - private_attribute
    + public_method()
    - private_method()
}
```

- **+** indicates public visibility
- **-** indicates private visibility
- Methods include parentheses to distinguish from attributes

Relationships

1. Inheritance (<|--):

```
Config <|-- ImagePreprocessor
```

- The empty triangle points to the parent class
- The line connects to the child class
- Represents "is-a" relationship

2. Composition (*--):

```
ImagePreprocessor *-- FaceDetector
```

- The filled diamond is on the containing class
- Represents "has-a" relationship with lifecycle dependency

3. Dependency (..>):

```
ImagePreprocessor ..> Utils
```

- Dotted line with arrow
- Represents "uses" relationship

Special Notations

1. Abstract Classes:

```
abstract class Config
```

- Indicates class cannot be instantiated directly

2. Utility Classes:

```
class Utils <<utility>>
```

- Stereotype notation indicates utility/helper class
- Contains static methods only

3. Notes:

```
note right of ClassName  
    Note text here  
end note
```

- Adds explanatory text to diagram
- Can be positioned left/right/top/bottom of elements

Package Organization

```
package "Package Name" {  
    // Classes and relationships  
}
```

- Groups related classes together
- Improves diagram organization and readability

Key Differences from Mermaid

1. PlantUML uses `@startuml/@enduml` tags
2. Relationship syntax is different (`<|--` vs `-->`)
3. More detailed stereotypes and visibility modifiers
4. Package organization capabilities
5. More flexible note placement options

This PlantUML diagram provides a more detailed and technically accurate representation of the Python module structure, including proper typing information and visibility modifiers that weren't shown in the Mermaid version.