## Project Documentation: Naming Conventions

**Project Overview**

This document outlines the naming conventions that will be used throughout the CNN project. Adhering to these conventions ensures that the codebase is consistent, readable, and maintainable. These standards will be applied to all aspects of the project, including classes, functions, variables, files, and folders.

**1. Naming Conventions**

**1.1 Class Names**

* **Convention**: PascalCase
* **Description**: Class names will be written in PascalCase, where the first letter of each word is capitalized without any underscores or spaces.
* **Example**:
  + ImageProcessor
  + ModelTrainer
  + DataLoader

**1.2 Function Names**

* **Convention**: camelCase
* **Description**: Function names will be written in camelCase, where the first letter of the first word is lowercase, and the first letter of each subsequent word is capitalized.
* **Example**:
  + loadData()
  + trainModel()
  + evaluatePerformance()

**1.3 Variable Names**

* **Convention**: lowercase
* **Description**: Variable names will follow the lowercase convention, similar to function names, to maintain consistency across the codebase.
* **Example**:
  + imageData
  + trainingSet
  + learningRate

**1.4 Constant Names**

* **Convention**: UPPER\_SNAKE\_CASE
* **Description**: Constants will be named using uppercase letters with words separated by underscores.
* **Example**:
  + MAX\_ITERATIONS
  + LEARNING\_RATE
  + BATCH\_SIZE

**1.5 File Names**

* **Convention**: snake\_case
* **Description**: File names will be written in lowercase with words separated by underscores. This ensures compatibility across different operating systems.
* **Example**:
  + data\_loader.py
  + model\_trainer.py
  + image\_processor.py

**1.6 Folder Names**

* **Convention**: kebab-case
* **Description**: Folder names will be written in lowercase with words separated by hyphens. This naming convention is chosen for its readability and ease of use in command-line environments.
* **Example**:
  + data-processing
  + model-checkpoints
  + trained-models

**4. Conclusion**

Following these naming conventions will help maintain a clean, organized, and professional codebase throughout the CNN project. This documentation serves as a reference to ensure all team members and contributors adhere to the same standards, promoting consistency and ease of collaboration.