**INSTRUCTIONS**

**Introduction:**

Image classification is a task in computer vision where the goal is to assign a label or category to an input image based on its content. It is a type of supervised learning, where the algorithm is trained on a dataset of labeled images to learn patterns and features associated with different classes.

**Language & Framework:**

Python and Jupyter Notebook have been used for this Image Classification task.

**Packages & Libraries Used:**

OpenCV, Numpy, Scikit-Learn, TensorFlow, Matplotlib & Seaborn.

**Loading & Preprocessing:**

All five classes of folders have been used as sub-folders and loaded in the Jupyter Notebook files. From there it was loaded and images were preprocessed. All the images were read, resized, labeled, and saved for each class. Then all the images were combined and simultaneous labels were also combined and stored in the Numpy array. For simplifying the handling of labels, especially in scenarios like one-hot encoding. All the images’ pixels were normalized in the range from 0 to 1.

**Training Model:**

After combining and normalizing the images, it is split into two sets which are train and test datasets. Since there are more Machine Learning algorithms for classification tasks, I chose the Convolutional Neural Network model from deep learning to carry out this task because it is the model widely used nowadays for handling images. A Convolutional Neural Network (CNN) is a specialized type of deep neural network designed for visual data processing, particularly in tasks like image classification and object recognition. Its architecture comprises layers such as convolutional layers, which apply filters to detect spatial patterns, pooling layers for spatial down-sampling, and fully connected layers for high-level reasoning and classification. CNNs excel at automatically learning hierarchical representations of features in images, allowing them to capture both local details and global structures. With concepts like transfer learning and regularization techniques such as dropout, CNNs have become a cornerstone in computer vision, demonstrating remarkable performance in diverse visual recognition tasks.

**Model Evaluation:**

In classification tasks, model evaluation involves assessing how well a machine learning model performs in assigning class labels to instances. Common metrics include accuracy, measuring overall correctness; precision, emphasizing the accuracy of positive predictions; recall, capturing the model's ability to identify all relevant instances; and the F1 score, which balances precision and recall. The confusion matrix provides a detailed breakdown of true positives, true negatives, false positives, and false negatives. All these metrics have been used to evaluate the model.

**Questions**

**How did you solve the problem?**

I briefly explained the process on the previous page and the source code also be attached. Problem statement, Data Preparation, Model Training, Model Testing, Model Evaluation, and Hyperparameter Tuning are the overview of this task.

**Why did you choose to solve the problem the way you did?**

As I explained in the Instruction part, CNN is the best to carry out the tasks from images. Also, I have done projects using different ML algorithms but I haven’t done any projects from the CNN model so I chose that model from Deep learning. Then I wanted to do this project as soon as possible so I used only an image dataset. If I have some more time, I will try to sort out the task using both image and OCR text which will increase the performance of the model.

**What challenges did you face?**

Initially, It was hard to load the folders of images, While preprocessing, the model wasn’t reading the images and confusing in class names so it was labeled efficiently, Hyperparameter Tuning took so much time to find appropriate combinations, and I could’t have enough infrastructure in my laptop to do my best.

**If you had more time, what would you do differently?**

More hyperparameters will be used, Random search or Grid search have been used for finding good combinations of hyperparameters. More parameters are there in the CNN that can be incorporated. Transfer learning could also be an option for better results. Different models were used along with both image and OCR datasets. If I had enough infrastructures like a cloud platform. I’ll perform different combinations in many nodes will save time and allow me to do the task as soon as possible. There are so many ways to do tasks I can do them in many ways and if I can’t do them, definitely I’ll learn and do that without quitting.