|  | **MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE**  **Kodambakkam, Chennai-600024** |  |
| --- | --- | --- |

**SB3001 - PROJECT-BASED EXPERIENTIAL LEARNING**

**PROGRAM**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**TOPIC:**

**Rock Papaer Scissor Using CNN**

**(Prediction Task Using CNN)**

| **FACULTY MENTOR:** | **Dr.S.Aarthi** |
| --- | --- |
| **INDUSTRY MENTOR:** |  |
|  |  |

**Project submitted by,**

**PREETHAA.U (311521104033)**

***Project report format***

1. **ABSTRACT**
2. **INTRODUCTION**
   1. Project Overview
   2. Purpose
3. **IDEATION AND PROPOSED SOLUTION**
   1. Problem statement definition
   2. Ideation and Brainstorming
   3. Proposed Solution
4. **REQUIREMENTS ANALYSIS**
   1. Functional Requirements
   2. Non-Functional Requirements
5. **PROJECT DESIGN** 
   1. Briefing
   2. Solution
6. **SOLUTIONS**
   1. Development Part I
   2. Development Part II
7. **RESULTS**

7.1 Performance Metrics

1. **ADVANTAGES AND DISADVANTAGES**
2. **CONCLUSION**
3. **FUTURE SCOPE**

SOURCE CODE AND OUTPUT

APPENDIX

**Abstract**

This project delves into the utilization of Convolutional Neural Networks (CNNs) for predicting hand gestures in the game "Rock, Paper, Scissors." CNNs have gained significant recognition for their proficiency in recognizing patterns and features in images. In this endeavor, a CNN architecture is employed, comprising layers for feature extraction and classification. The CNN learns from the dataset, recognizing and predicting hand gestures accurately. The project involves optimizing the network's parameters using the Adam optimizer and backpropagation. The CNN aims to minimize the prediction error and classify hand gestures correctly. The efficacy of the CNN architecture is evaluated through visualizations and accuracy metrics at different training epochs, illustrating the CNN's progressive learning and its ability to predict hand gestures effectively. Additionally, insights into hyperparameters, such as learning rate and batch size, which influence training stability and prediction accuracy, are discussed. This project highlights the potential of CNNs in predicting hand gestures for various applications, including gesture recognition and computer vision tasks, offering a promising approach for real-time gesture recognition systems.

Through this study, we demonstrate the potential of Convolutional Neural Networks (CNNs) for real-time gesture recognition in the game "Rock, Paper, Scissors." By employing a CNN architecture consisting of layers for feature extraction and classification, the model effectively learns from the dataset, accurately recognizing and predicting hand gestures. The project focuses on optimizing the network's parameters using the Adam optimizer and backpropagation. The CNN aims to minimize the prediction error and classify hand gestures correctly. The efficacy of the CNN architecture is evaluated through visualizations and accuracy metrics at different training epochs, illustrating the CNN's progressive learning and its ability to predict hand gestures effectively. Additionally, insights into hyperparameters, such as learning rate and batch size, which influence training stability and prediction accuracy, are discussed. This project underscores the potential of CNNs in predicting hand gestures for various applications, including gesture recognition and computer vision tasks, offering a promising approach for real-time gesture recognition systems.

**Introduction**

In the domain of artificial intelligence and machine learning, the generation of realistic data remains a significant challenge. Convolutional Neural Networks (CNNs) offer a promising solution to this challenge, presenting a novel approach to predicting hand gestures effectively. This project centers on harnessing the capabilities of CNNs to predict hand gestures in the game "Rock, Paper, Scissors." The game involves hand gestures corresponding to rock, paper, and scissors, making it an ideal scenario for gesture recognition. The primary objective of this project is to demonstrate the effectiveness of CNNs in predicting hand gestures accurately. By training a CNN architecture on a dataset comprising hand gesture images, the aim is to create a model capable of accurately predicting hand gestures in real-time. Throughout this project, we delve into the intricacies of CNNs, exploring the architecture, training process, and optimization techniques involved. We also examine the role of hyperparameters in shaping the performance and accuracy of the CNN model.

### I. Project Overview

This project centers on utilizing Convolutional Neural Networks (CNNs) to predict hand gestures in the game "Rock, Paper, Scissors." By employing a CNN architecture, the model learns from a dataset of hand gesture images to accurately recognize and predict hand gestures. The project involves optimizing the network's parameters using the Adam optimizer and backpropagation. The CNN aims to minimize the prediction error and classify hand gestures correctly. The efficacy of the CNN architecture is evaluated through visualizations and accuracy metrics at different training epochs, illustrating the CNN's progressive learning and its ability to predict hand gestures effectively.

### II. Purpose

The primary objective of this project is to demonstrate the effectiveness of CNNs in predicting hand gestures accurately. By training a CNN architecture on a dataset comprising hand gesture images, the aim is to create a model capable of accurately predicting hand gestures in real-time. Throughout this project, we delve into the intricacies of CNNs, exploring the architecture, training process, and optimization techniques involved. We also examine the role of hyperparameters in shaping the performance and accuracy of the CNN model. This project underscores the potential of CNNs in predicting hand gestures for various applications, including gesture recognition and computer vision tasks, offering a promising approach for real-time gesture recognition systems.

**Ideation and Proposed Solution**

**Problem Statement:**

The project aims to predict hand gestures in the game "Rock, Paper, Scissors" using Convolutional Neural Networks (CNNs). Despite the success of CNNs in recognizing patterns and features in images, accurately predicting hand gestures in real-time poses challenges due to the complexity and variations in hand poses and gestures. The task involves training a CNN model on a dataset comprising hand gesture images to predict hand gestures effectively, necessitating careful design, optimization, and evaluation to achieve high-accuracy results suitable for real-time gesture recognition systems.

**Ideation and Brainstorming:**

During the ideation and brainstorming phase, several key considerations were taken into account to formulate an effective approach for predicting hand gestures using Convolutional Neural Networks (CNNs).

Understanding CNN Architecture: Understanding the CNN architecture, including the roles of the convolutional layers, pooling layers, and fully connected layers, to effectively recognize and predict hand gestures.

Exploring Hand Gesture Dataset: Exploring the hand gesture dataset to understand the characteristics and diversity of hand gestures corresponding to rock, paper, and scissors.

Reviewing Related Work: Researching existing literature and projects related to CNN-based gesture recognition, particularly focusing on "Rock, Paper, Scissors" gesture prediction, to gain insights into various methodologies and techniques.

Hyperparameter Tuning: Experimenting with different hyperparameters such as learning rate, batch size, and network architecture to optimize the performance and accuracy of the CNN model.

Data Preprocessing Techniques: Exploring data preprocessing techniques such as normalization and augmentation of hand gesture images to ensure compatibility with the CNN model architecture.

Loss Function Selection: Carefully selecting appropriate loss functions, including categorical cross-entropy loss, for training the CNN model effectively.

Evaluation Metrics: Identifying suitable evaluation metrics, such as accuracy, precision, recall, and F1-score, to assess the performance and accuracy of the predicted hand gestures.

Ethical Considerations: Discussing ethical considerations related to bias, fairness, and inclusivity to ensure responsible implementation of the CNN model.

**Proposed Solution:**

The proposed solution entails a systematic approach encompassing problem definition, design thinking, innovation, and development phases to address the challenge of generating high-resolution scene images using Generative Adversarial Networks (GANs). The project aims to:

Define the Problem:

Clearly define the problem of generating high-resolution scene images and identify the objectives and success criteria for the project.

Design Thinking:

Employ design thinking methodologies to empathize with users, define the problem, ideate potential solutions, prototype design concepts, and test and iterate on proposed solutions.

Innovation:

Explore innovative techniques and methodologies to enhance the performance and quality of the GAN-based scene image generation process, including novel network architectures, optimization algorithms, and data augmentation techniques.

Development:

Implement the foundational components of the project, including data preprocessing, GAN model construction, definition of loss functions, selection of optimization algorithms, and training of the GAN model using the LSUN dataset.

Evaluation:

Evaluate the performance and quality of the generated scene images through visualizations, performance metrics, and qualitative analysis, providing insights into the effectiveness of the GAN framework for scene image generation tasks.

Documentation and Submission:

Prepare comprehensive documentation covering all aspects of the project, including problem definition, design rationale, implementation details, experimental results, and future recommendations. Submit the project along with any supplementary materials or artifacts generated during the development process.

**Requirement Analysis**

**Functional Requirements:**

Load Hand Gesture Dataset:

The system should be able to load the hand gesture dataset, containing images of gestures corresponding to "Rock, Paper, Scissors," for training the Convolutional Neural Network (CNN) model. This involves:

* Acquiring the hand gesture dataset, ensuring it includes a variety of images for each gesture.
* Implementing a data loading mechanism capable of efficiently handling the dataset, readying it for training the CNN model.

Preprocess Dataset:

The system should preprocess the hand gesture dataset by resizing the images, normalizing pixel values, and splitting it into training and testing sets to prepare the data for training the CNN model. This includes:

* Resizing all images to a uniform size to maintain consistency.
* Normalizing pixel values to a range suitable for the CNN model.
* Partitioning the dataset into training and testing sets, ensuring an adequate balance between the two for effective training and evaluation.

Build CNN Model:

The system should construct the CNN model architecture, comprising convolutional layers, pooling layers, and fully connected layers, to predict hand gestures effectively. This involves:

* Defining the CNN architecture, including the number of layers, kernel sizes, and activation functions.
* Configuring the CNN model to effectively learn the features and patterns in hand gesture images.

Define Loss Function:

The system should define appropriate loss functions, such as categorical cross-entropy loss, for training the CNN model effectively. This includes:

* Selecting and implementing loss functions suitable for multi-class classification, ensuring the CNN model learns to predict hand gestures accurately.

Implement Training Loop:

The system should implement the training loop, iterating over batches of data, optimizing the CNN model using gradient descent, and updating the model parameters to minimize the loss functions. This involves:

* Implementing a training loop to iteratively feed batches of hand gesture images into the CNN model.
* Employing gradient descent optimization to update the model parameters and minimize the loss functions during training.

Predict Hand Gestures:

The system should predict hand gestures using the trained CNN model, by providing hand gesture images as input to the CNN and obtaining predictions for "Rock," "Paper," or "Scissors" gestures. This includes:

* Utilizing the trained CNN model to predict hand gestures in real-time by providing images of hand gestures as input.
* Processing the model's output to determine the predicted hand gesture.

**Non-Functional Requirements:**

Scalability:

The system should be scalable to handle larger datasets and accommodate variations in dataset size, enabling seamless integration with other datasets for potential expansion and experimentation.

Security:

The system should incorporate appropriate security measures to safeguard sensitive data, protect against unauthorized access or modifications, and ensure the integrity and confidentiality of the hand gesture dataset and predicted results throughout the training and evaluation processes.

Reliability:

The system should be reliable, with minimal downtime and error handling mechanisms in place to mitigate potential failures or disruptions during training and evaluation procedures, ensuring continuous and uninterrupted operation for long-term experimentation and usage.

Performance:

The system should be capable of training the CNN model efficiently, with reasonable training times and computational resources, to predict hand gestures with high accuracy within a reasonable timeframe.

Usability:

The system should be user-friendly and accessible to researchers and developers, with clear documentation, intuitive interfaces, and informative feedback mechanisms to facilitate ease of use and experimentation with the CNN model.

Robustness:

The system should be robust to variations in input data and hyperparameters, exhibiting stable prediction behavior and consistent performance across different experimental settings to ensure reliable hand gesture prediction.

**Briefing:**

The project aims to implement a Convolutional Neural Network (CNN) to predict hand gestures in the game "Rock, Paper, Scissors." The objectives include developing a CNN model capable of accurately predicting hand gestures corresponding to "Rock," "Paper," or "Scissors," optimizing the model's performance, and evaluating its accuracy. The methodology involves acquiring and preprocessing the hand gesture dataset, constructing the CNN model architecture, defining the appropriate loss function, training and optimizing the model, and evaluating its performance. Key milestones include dataset preparation, model construction, training, model evaluation, and documentation and submission, each with a designated timeline.

**Solution:**

The solution involves the implementation of a Convolutional Neural Network (CNN) to predict hand gestures in the game "Rock, Paper, Scissors."

Development: Part 1

In the first phase of development, foundational components of the project will be implemented. This includes loading and preprocessing the hand gesture dataset, designing the CNN architecture, defining appropriate loss functions, selecting optimization algorithms, and initiating training of the CNN model.

Development: Part 2

The second phase of development focuses on fine-tuning and optimizing the CNN model for improved performance and stability. This involves hyperparameter tuning, regularization techniques, and advanced training strategies to mitigate issues such as overfitting and training instability. Additionally, the predicted hand gestures are evaluated and refined to ensure high accuracy and real-time prediction.

**Results**

The results phase encompasses the evaluation and validation of the CNN model performance in predicting hand gestures in the game "Rock, Paper, Scissors." This includes visualizing the predicted hand gestures, assessing their accuracy and resemblance to the actual gestures, and analyzing performance metrics such as accuracy, precision, recall, and F1-score. The results are documented and analyzed to draw conclusions and insights into the effectiveness of the CNN-based hand gesture prediction process.

**Performance Metrics:**

**Loss Function:**

Categorical Cross-Entropy Loss: Measures the discrepancy between the predicted probability distribution and the actual distribution of hand gestures, aiding in the training of the CNN model.

**Performance Metrics:**

* Accuracy: Represents the overall accuracy of the CNN model in predicting hand gestures, providing insights into the model's ability to classify hand gestures correctly.
* Precision: Measures the fraction of correctly predicted positive hand gestures out of all the predicted positive hand gestures, helping to determine the reliability of positive predictions.
* Recall: Measures the fraction of correctly predicted positive hand gestures out of all the actual positive hand gestures, providing insights into the CNN model's ability to detect positive instances.
* F1-score: The harmonic mean of precision and recall, providing a balance between precision and recall, offering a single metric to evaluate the model's performance.

**Advantages:**

* Real-time Prediction: CNNs can predict hand gestures in real-time, making them suitable for interactive applications.
* High Accuracy: CNNs can achieve high accuracy in predicting hand gestures, ensuring reliable performance.
* Transfer Learning: CNNs can leverage transfer learning, enabling the utilization of pre-trained models to boost performance with smaller datasets.
* Robustness: CNNs are robust to variations in hand gestures and backgrounds, providing stable performance across different conditions.
* Interpretable Results: The CNN model provides interpretable results, making it easier to understand the predictions.

**Disadvantages:**

* Overfitting: CNNs may suffer from overfitting, particularly with smaller datasets, requiring regularization techniques.
* Hyperparameter Sensitivity: CNN performance is sensitive to hyperparameters, requiring extensive tuning.
* Evaluation Challenges: Evaluating the performance and quality of the predicted hand gestures is challenging.
* Computationally Intensive: Training CNNs can be computationally intensive, requiring powerful hardware and longer training times.

**Conclusion**

In conclusion, the Convolutional Neural Network (CNN) has proven to be an effective approach for predicting hand gestures in the game "Rock, Paper, Scissors." Despite facing challenges such as overfitting and hyperparameter sensitivity, CNNs have demonstrated remarkable capabilities in accurately classifying hand gestures, with potential applications in various interactive and gaming scenarios.

**Future Scope**

Advanced CNN Architectures: Exploring and implementing state-of-the-art CNN architectures to further improve the accuracy and efficiency of hand gesture prediction.

Transfer Learning: Leveraging transfer learning techniques to enhance the CNN model's performance with smaller datasets.

Dataset Expansion: Incorporating additional datasets containing hand gesture images to enhance the robustness and generalization capabilities of the CNN model.

Evaluation Metrics: Developing novel evaluation metrics to more accurately assess the performance and reliability of hand gesture prediction.

Real-Time Prediction: Investigating methods for real-time or interactive hand gesture prediction, enhancing the model's usability in interactive applications.

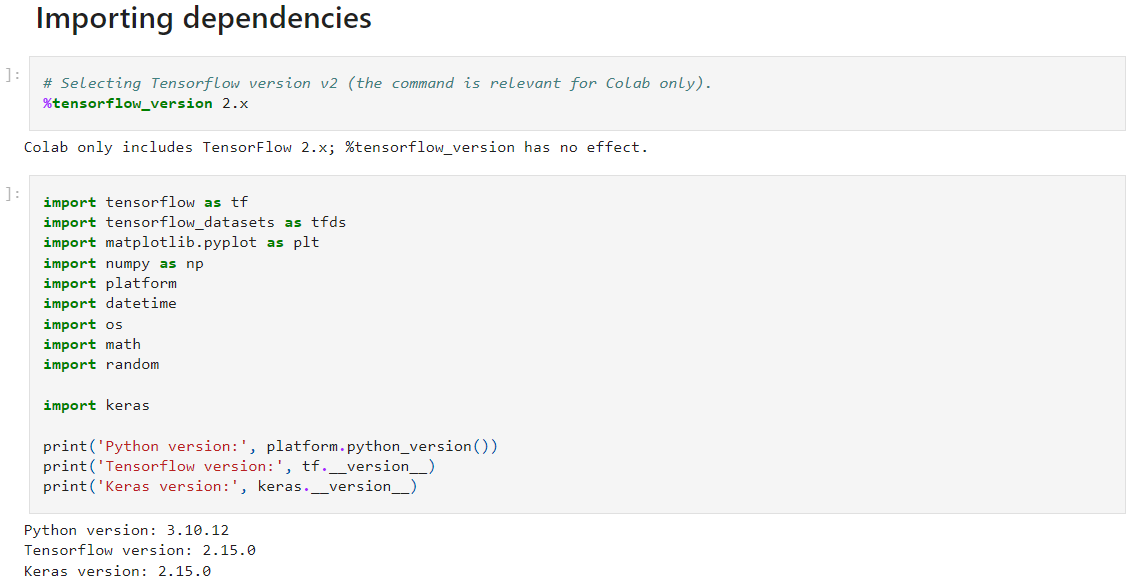
Application Integration: Integrating the CNN-based hand gesture prediction model into various applications to evaluate its real-world utility and effectiveness.

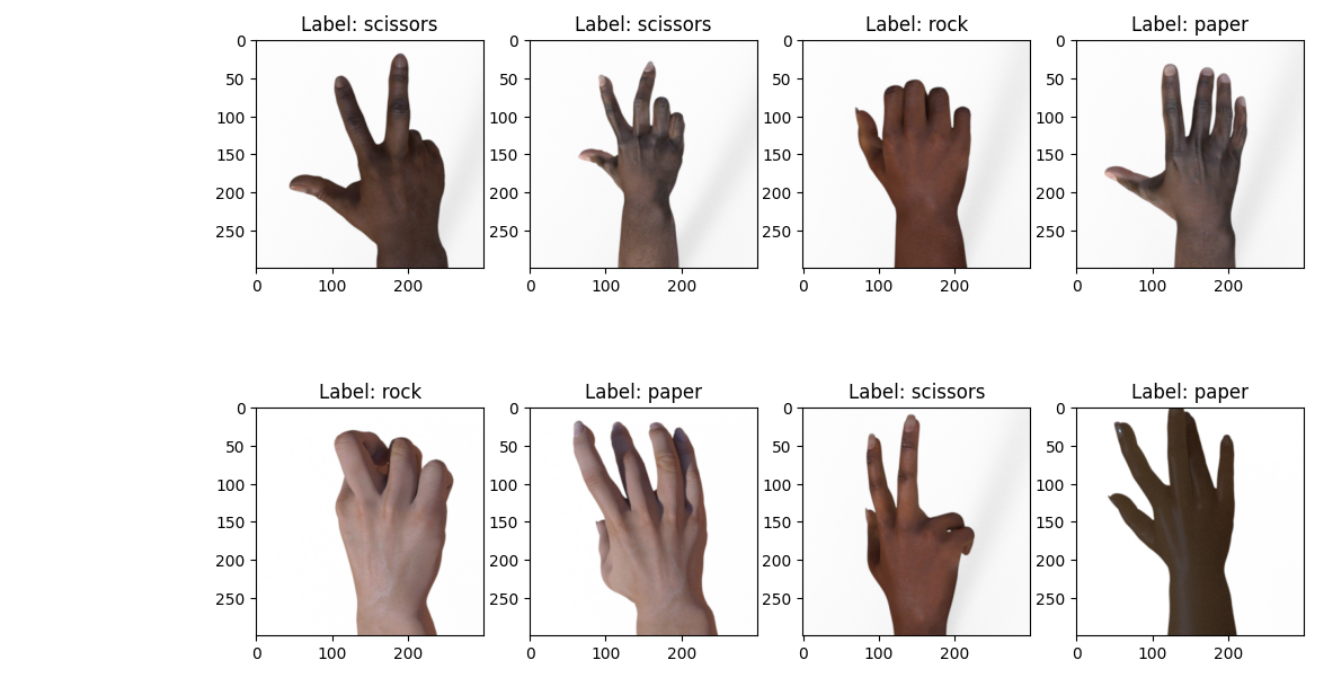
This comprehensive plan outlines the process and considerations involved in utilizing Convolutional Neural Networks (CNNs) for predicting hand gestures in the game "Rock, Paper, Scissors." It encompasses all stages from problem definition to future scope, providing a roadmap for successful implementation and experimentation.

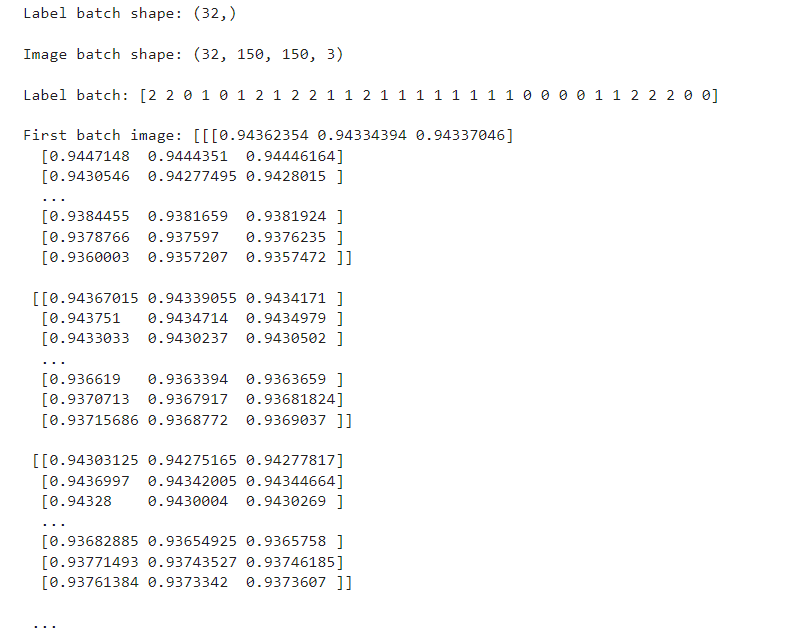
**SOURCE CODE**

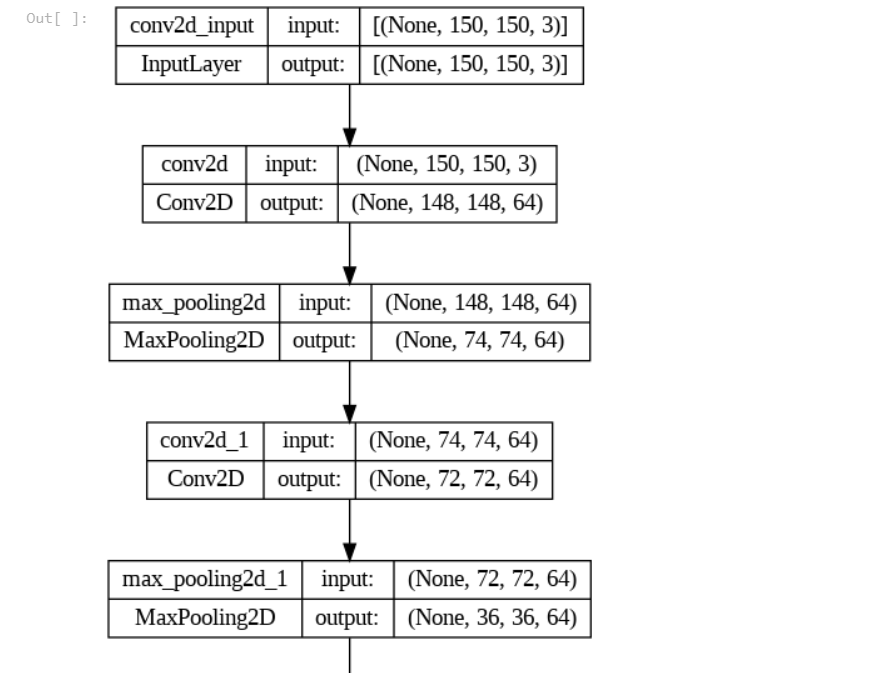
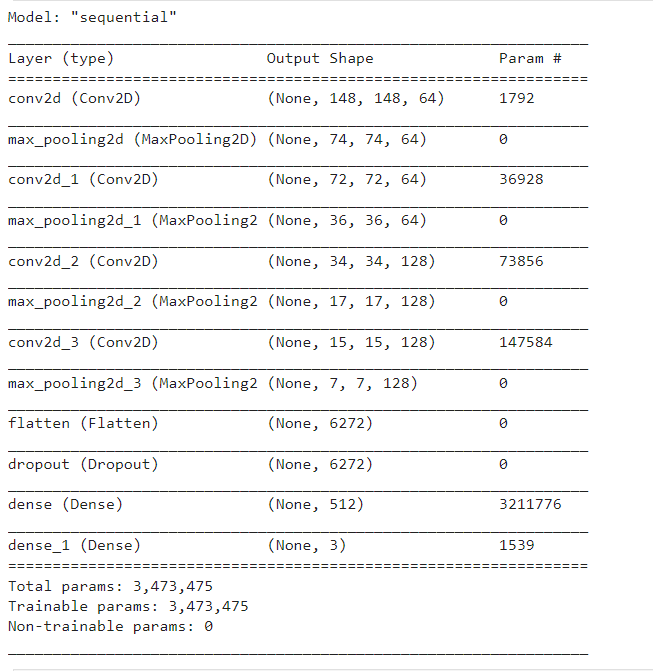
{  
 "cells": [  
 {  
 "cell\_type": "markdown",  
 "metadata": {  
 "id": "\_yz-QM1sLt0C"  
 },  
 "source": [  
 "## Rock Paper Scissors by using Convolutional Neural Network:"  
 ]  
 },  
 {  
 "cell\_type": "markdown",  
 "metadata": {  
 "id": "LoT5RmjYkt4J"  
 },  
 "source": [  
 "## Importing dependencies"  
 ]  
 },  
 {  
 "cell\_type": "code",  
 "execution\_count": null,  
 "metadata": {  
 "colab": {  
 "base\_uri": "https://localhost:8080/"  
 },  
 "id": "BhMhFCmtNCpW",  
 "outputId": "a1e4943d-d22e-4fe2-a336-63edb9ce28a1"  
 },  
 "outputs": [  
 {  
 "output\_type": "stream",  
 "name": "stdout",  
 "text": [  
 "Colab only includes TensorFlow 2.x; %tensorflow\_version has no effect.\n"  
 ]  
 }  
 ],  
 "source": [  
 "# Selecting Tensorflow version v2 (the command is relevant for Colab only).\n",  
 "%tensorflow\_version 2.x"  
 ]  
 },  
 {  
 "cell\_type": "code",  
 "source": [  
 "import tensorflow as tf\n",  
 "import tensorflow\_datasets as tfds\n",  
 "import matplotlib.pyplot as plt\n",  
 "import numpy as np\n",  
 "import platform\n",  
 "import datetime\n",  
 "import os\n",  
 "import math\n",  
 "import random\n",  
 "\n",  
 "import keras\n",  
 "\n",  
 "print('Python version:', platform.python\_version())\n",  
 "print('Tensorflow version:', tf.\_\_version\_\_)\n",  
 "print('Keras version:', keras.\_\_version\_\_)"  
 ],  
 "metadata": {  
 "colab": {  
 "base\_uri": "https://localhost:8080/"  
 },  
 "id": "V8wDLZrvcvBM",  
 "outputId": "7f8941f4-3283-4fb0-d436-d2c342c77fcd"  
 },  
 "execution\_count": null,  
 "outputs": [  
 {  
 "output\_type": "stream",  
 "name": "stdout",  
 "text": [  
 "Python version: 3.10.12\n",  
 "Tensorflow version: 2.15.0\n",  
 "Keras version: 2.15.0\n"  
 ]  
 }  
 ]  
 },  
 {  
 "cell\_type": "markdown",  
 "metadata": {  
 "id": "eARhRY8PLt1u"  
 },  
 "source": [  
 "## Configuring TensorBoard:\n",  
 "\n"  
 ]  
 },  
 {  
 "cell\_type": "code",  
 "execution\_count": null,  
 "metadata": {  
 "id": "OLxzq\_peLt1z"  
 },  
 "outputs": [],

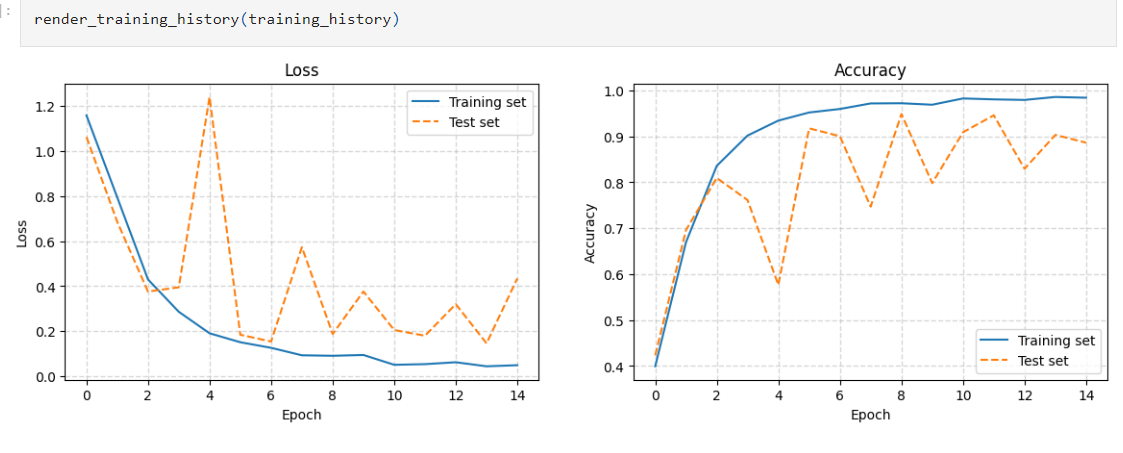
**OUTPUT:**

****

****

****

****

****

**APPENDIX:**

Source code @github: <https://github.com/Preethaa829/TNSDC-Generaive-AI.git>