# RPS_PROJECT

## Table of Contents

## 1. Introduction

### 1.1 Problem Statement

This project tackles automatic recognition of the three gestures in Rock–Paper–Scissors “rock, paper, and scissors” from both still images and a live webcam feed. Beyond simply labeling a frame, the goal is to handle the messy parts of reality: variable lighting, mixed or green-screen backgrounds, different hand orientations and sizes, and the speed requirements of interactive use on a Mac laptop. The full solution therefore spans data cleaning, model training, and real-time inference, not just a standalone classifier.

## 1.2 Objectives

Our aim was to build an end-to-end pipeline that we could trust. We began by organizing and cleaning a public RPS dataset, removing green backgrounds and softening edges to reduce segmentation artifacts. We then designed and trained three convolutional neural networks of increasing complexity, keeping the input size fixed to make comparisons fair. A central target was to reach reliable validation accuracy around or above 90% on held out images, but stability mattered as much as peak numbers: smooth learning curves and consistent behavior across runs were essential. Finally, we wanted the best model to transfer cleanly to real time, so we integrated MediaPipe to locate the hand, OpenCV to stream frames, and TensorFlow-Metal to keep inference responsive on macOS.

## 1.3 Constraints and Limitations

The dataset is modest and slightly imbalanced, which can bias learning if not treated carefully. Many images were captured on a green backdrop while others were cluttered; this mismatch can confuse a model that overfits to background cues, so we standardized with green screen removal, white background replacement, and edge smoothing. Hardware also imposes limits: training and inference run on a Mac with Apple's Metal backend, which encourages compact architectures and efficient input pipelines. In the live setting we depend on MediaPipe's keypoints to crop the hand region; when detections are off (fast motion, partial hands, unusual poses), predictions can degrade. And, as with any vision model trained on a finite group of users and environments, distribution shifts—skin tones, accessories, camera angles, or lighting not seen in training—may reduce accuracy.

## 1.4 Overview of Model 1, 2, and 3

- **Model 1 (baseline, compact CNN).** A small sequential network with three convolutional blocks followed by a dense head, trained on 200×300 RGB inputs with moderate augmentation. In our runs it produced the most stable validation curves and the best overall validation accuracy, while remaining fast for real-time use.

- **Model 2 (deeper capacity + regularization).** A slightly larger CNN that adds depth and dropout to capture more complex patterns. It occasionally improved training accuracy but proved more sensitive to overfitting and required tighter tuning to keep validation performance consistent.
- **Model 3 (pipeline-heavy + scheduling).** The same input resolution paired with a stronger preprocessing stage (green removal with softened edges), a quick batch-size sweep, class weighting, and learning rate scheduling. It demonstrated solid learning but showed volatile validation behavior on some splits, suggesting sensitivity to sampling and optimization choices.

## 2. Dataset Preparation

### 2.1 Raw dataset and split strategy

The dataset contains images arranged in three folders, one per class: paper, rock, scissors. Before splitting, corrupted files and duplicates are detected and removed. A stratified split of about 80–20 between training and validation is applied with a fixed seed, preserving class proportions in both sets and ensuring reproducibility.

### 2.2 Cleaning pipeline

The goal is to produce consistent images with the hand isolated and the context standardized.

**Green screen removal.** When a green background is present, processing is performed in HSV space with a threshold in a typical green range. The mask is refined with light morphological operations to suppress speckles and fill small holes. The background is replaced with uniform white to standardize the context.

**Edge softening.** To avoid harsh hand–background transitions, the mask is gently feathered over a few pixels, reducing halos and segmentation artifacts.

**Noise reduction.** A mild denoising filter is applied on the hand region to attenuate high-frequency noise while preserving finger lines and knuckles.

**Resizing and normalization.** All images are resized to 300×200 pixels and scaled to the [0, 1] range. This resolution balances visual quality with training and inference speed.

Two “before and after” figures are included in the report to document the effects of segmentation, feathering, and standardization.

## 2.3 Augmentation policy

Augmentation is applied only during training to increase variety without altering the semantic gesture.

- horizontal flip with probability 0.5
- small random rotations (about ±15–25 degrees)
- moderate translations and zoom (translations up to ~8%, zoom 0.9–1.1)
- light brightness and contrast jitter (about ±20–40%)
- optional mild Gaussian noise ($\sigma \approx 0.01$–$0.03$ in normalized units)

Ranges are chosen to avoid unrealistic poses and to keep the gesture’s salient features invariant.

## 2.4 Quality checks

Several quick quality checks are performed after cleaning and splitting.

**Per-class counts.** Class balance across paper, rock, and scissors is verified. When moderate imbalance is detected, class weights are used during training to compensate.

**Inspection grids.** Sample grids from the training set are reviewed to check hand centering, segmentation quality, and the absence of strong halos on the white background.

**Outliers.** Atypical shots are identified and excluded (noncanonical poses, heavy occlusions, extreme exposures).

**File integrity.** All files are confirmed readable and to match the expected dimensions after resizing.