Building a CNN for Rock-Paper-Scissors Classification

Kinda Kutkut

June 2025

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

Abstract

This report addresses the task of classifying hand gestures from the Rock-Paper-Scissors game using Convolutional Neural Networks (CNNs). The objective is to develop a model that can accurately identify one of three gestures (rock, paper, or scissors) from an RGB image. Three CNN architectures with increasing complexity were designed and trained. Hyperparameter tuning was applied to the first model using a grid search strategy. All models were evaluated on a held-out test set using accuracy, precision, recall, F1-score, and confusion matrix analysis. The best-performing model achieved a test accuracy of 98.48%. A comparative analysis of the models illustrates the impact of architectural depth and regularization on classification performance and generalization.

Introduction

The classification of hand gestures is a fundamental task in computer vision, with applications in gesture recognition, human-computer interaction, and game-based systems. This report focuses on the specific case of recognizing gestures from the Rock-Paper-Scissors game using image-based classification methods. Each gesture corresponds to a distinct hand pose, and the objective is to automatically classify an input image as either rock, paper, or scissors.

Convolutional Neural Networks (CNNs) are particularly well-suited for image-based tasks due to their ability to extract spatial hierarchies of features [1]. In this project, three CNN architectures of increasing complexity were designed and evaluated. The aim was to assess how model depth affects classification performance.

All models were trained using the TensorFlow and Keras frameworks, and the evaluation followed standard machine learning practices. Performance was measured using accuracy, precision, recall, and F1-score on a held-out test set. SEED = 42 was maintained throughout the project for all random procedures to insure reproducibility

Dataset Exploration

The dataset used in this project is the Rock-Paper-Scissors image dataset, published on Kaggle by Dr. G. Freeman[2]. The version accessed for this work was retrieved on June 20, 2025. It consists of 2,188 RGB images, each belonging to one of three classes: rock, paper, or scissors. These images represent hand gestures taken on a green background with consistent lighting and white balance, as noted in the dataset documentation.

Each image is of size 300×200 pixels and stored in PNG format. The dataset is organized into three folders—`rock`, `paper`, and `scissors`—corresponding to the gesture label. The class distribution is nearly balanced, with each category containing approximately 700 samples, ensuring that no particular gesture is overrepresented.

Visual inspection of random samples confirmed the presence of a uniform green background across most images, which is advantageous for model training as it reduces background-related noise. However, there is variability in the choice of the hand used for the gesture (left or right) which may impact the model's generalization performance.

Preprocessing

To ensure a statistically sound evaluation, the dataset was first split into training, validation, and test sets *before* any preprocessing was applied. This guarantees that no data manipulation depends on test set information, thereby preventing data leakage and preserving the integrity of the evaluation process.

The original images, sized at 300×200 pixels, were resized to 128×128 without preserving the aspect ratio. This resizing was chosen to reduce computational load and training time, while maintaining sufficient resolution for accurate gesture recognition. The images are well-lit, with a uniform background and clearly visible gestures, making them resilient to minor distortion introduced by resizing.

Pixel values were normalized to the $[0, 1]$ range by rescaling with a factor of 1/255, which accelerates training and improves numerical stability during gradient updates.

For the training set, data augmentation was applied using `ImageDataGenerator`. Augmentation operations included small random rotations, width and height shifts, zooming, and horizontal flipping. These transformations were limited to 10–15% to ensure label integrity while improving the model's ability to generalize[3]. Importantly, the validation and test sets were not augmented, as doing so would introduce randomness into model evaluation and potentially lead to misleading results.

Instead, both validation and test images were only normalized, reflecting how the model is expected to perform on real, unseen data. This separation ensures the test set remains a true proxy for deployment conditions.

Mini-batch training was employed with a batch size of 32. This approach balances noisy updates from single-image training and the inefficiency of full-batch training, allowing for stable and efficient gradient descent.

Model 1

Model Architecture

Model 1 is a simple convolutional neural network (CNN) designed to serve as a baseline for the Rock-Paper-Scissors classification task. It consists of two convolutional layers followed by max pooling, a fully connected dense layer, and a final softmax output layer.

The first convolutional layer uses 32 filters of size 3×3 with ReLU activation, followed by a max pooling layer with pool size 2×2. The second convolutional layer uses 64 filters of the same kernel size and is also followed by a 2×2 max pooling operation.

The feature maps are then flattened and passed through a dense layer with 64 units and ReLU activation. A dropout layer is applied after the dense layer to reduce overfitting by randomly disabling a fraction of neurons during training. The final output layer uses softmax activation with 3 units, corresponding to the three gesture classes.

This architecture is deliberately kept simple to ensure fast training and serve as a reference point for evaluating the impact of increased model complexity in subsequent models.

Hyperparameter Tuning

Hyperparameter tuning was performed for learning rate and dropout rate using a grid search strategy. The number of epochs was not tuned since early stopping was employed to terminate training once the validation loss ceased improving. The batch size was fixed during preprocessing and not included in the search space.

A total of nine combinations were tested across learning rates {0.01, 0.005, 0.001} and dropout rates {0.1, 0.3, 0.5}.

Each model was trained with the same architecture and evaluated using validation loss. The validation loss curves (Figure 1) showed that the combinations using a learning rate of 0.001 consistently outperformed higher values. The best-performing configuration used a learning rate of 0.001 and a dropout rate of 0.1. This model achieved the lowest validation loss and was selected as the final version of Model 1.

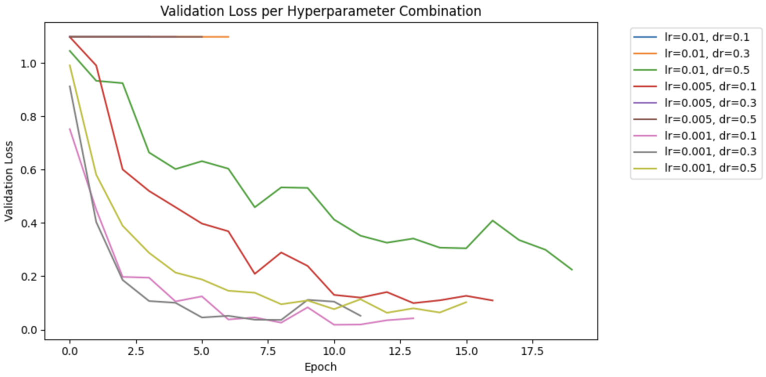

Figure 1: Validation loss curves for different learning rate and dropout combinations during Model 1 tuning.

Testing

The final model selected through hyperparameter tuning, using a learning rate of 0.001 and a dropout rate of 0.1, was evaluated on the held-out test set to estimate its generalization performance.

This model achieved a test accuracy of 97.87%, indicating strong performance on previously unseen data. A confusion matrix (Figure 2) was generated to analyze classification errors. Most examples were correctly classified, with only minor confusion between classes. These results confirm the effectiveness of the model in learning robust visual features from the training data.

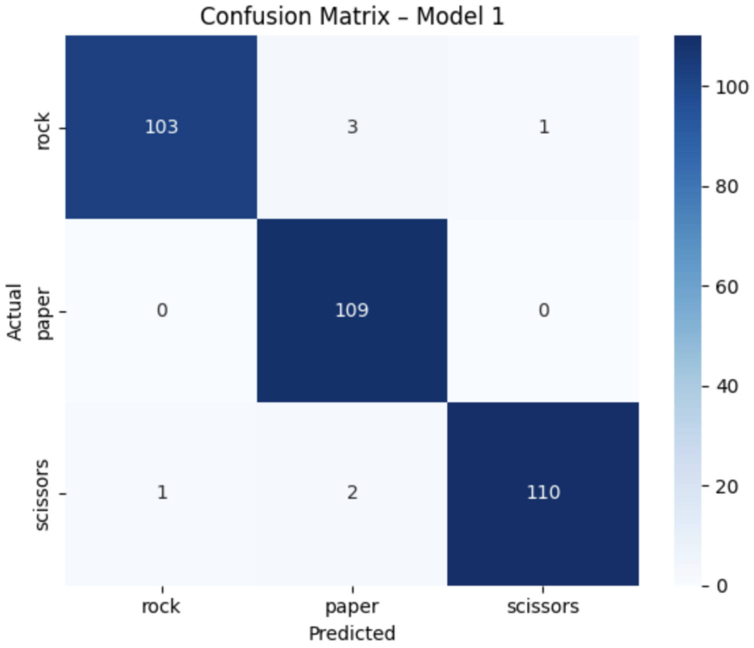

Figure 2: Confusion Matrix for Model 1

The classification report showed precision, recall, and F1-scores exceeding 96% across all three classes:

- **Rock:** Precision = 0.99, Recall = 0.96, F1-score = 0.98
- **Paper:** Precision = 0.96, Recall = 1.00, F1-score = 0.98
- **Scissors:** Precision = 0.99, Recall = 0.97, F1-score = 0.98

The model achieved an overall macro-averaged F1-score of 0.98, indicating consistent performance across all gesture categories.

Model 2

Model Architecture

Model 2 introduces a moderate increase in complexity compared to Model 1, incorporating three convolutional layers and a denser fully connected output structure. The model consists of the following components:

- A first convolutional layer with 32 filters of size 3×3, followed by a 2×2 max pooling layer.

- A second convolutional layer with 64 filters, also followed by a 2 × 2 max pooling layer.
- A third convolutional layer with 128 filters, followed by another max pooling operation.
- The feature maps are then flattened and passed through a dense layer with 128 units and ReLU activation.
- A dropout layer with a rate of 0.3 is applied to reduce overfitting.
- The final output layer consists of 3 units with softmax activation to produce class probabilities for the rock, paper, and scissors categories.

Compared to Model 1, this architecture balances additional depth and representational power with computational efficiency. The dense layer size was reduced to 128 units (from 512 in Model 1) to offset the increased convolutional depth, maintaining a reasonable training time.

Training

Model 2 was trained for a maximum of 20 epochs using the Adam optimizer and categorical crossentropy loss. Early stopping was employed with a patience of 3 to monitor the validation loss and prevent overfitting.

The training process showed steady improvements in both training and validation accuracy, with validation accuracy surpassing 96% after epoch 6. The training stabilized around epoch 11, where early stopping was triggered. The final recorded validation accuracy was 97.57%, with a corresponding validation loss of 0.0741.

This performance indicates that the increased convolutional depth in Model 2 helped extract more discriminative features without significantly increasing overfitting, as the gap between training and validation performance remained small throughout.

Testing

Model 2 was evaluated on the test set to assess its generalization performance. The test accuracy achieved was 0.9666 and Test Loss was 0.1147.

The classification report showed strong overall performance across all three classes, with a macro-averaged F1-score of 0.97. Precision and recall values were consistently high, especially for the "scissors" class, which achieved 1.00 precision and 0.98 recall. The "rock" and "paper" classes both attained F1-scores of 0.95.

The confusion matrix in Figure 3 highlights the model's ability to correctly classify most samples. A small number of misclassifications occurred between the "rock" and "paper" categories, but no samples were misclassified as "scissors." This indicates that the model was effective at distinguishing the most visually distinct class while occasionally confusing visually similar gestures.

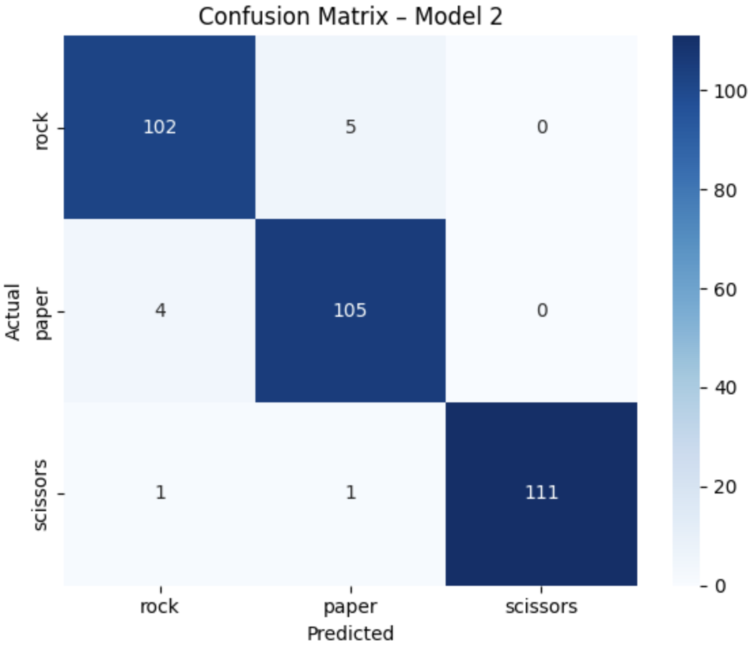

Figure 3: Confusion Matrix for Model 2

Model 3

Model Architecture

Model 3 introduces a deeper architecture to increase representational capacity. Compared to earlier models, it includes an additional convolutional layer with 128 filters, along with consistent use of `padding='same'` across all convolutional layers. This preserves spatial size through the convolution, which is especially helpful in deeper networks where fast image shrinkage is not desired, and also ensures the filter is applied to every region of the input equally, including the borders.

The network comprises four convolutional layers with increasing filter counts (32, 64, 128, 128), each followed by a 2 × 2 max pooling operation to progressively downsample feature maps. The final convolutional output is flattened and passed through a fully connected dense layer with 128 units and ReLU activation, followed by a dropout layer to reduce overfitting. The final output layer uses the softmax activation function to output class probabilities for the three gesture categories.

This deeper architecture is expected to perform better at distinguishing subtle variations in hand shapes, though at the potential cost of increased training time and susceptibility to overfitting if not regularized appropriately.

Training

Model 3 was trained using early stopping to monitor validation loss with a patience of three epochs. The architecture achieved rapid convergence within the first 5–6 epochs, consistently reaching validation accuracy above 97%. The best validation accuracy of 0.9878 was attained at multiple points during training, while the corresponding validation loss reached a minimum of 0.0266. These results suggest that deeper architectures with increased convolutional depth can improve representational capacity without overfitting, especially when regularized with dropout and monitored with early stopping.

Testing

Model 3 achieved strong performance on the test set with an overall accuracy of 0.9848 and a corresponding test loss of 0.0520. The classification report showed high precision and recall across all three classes, with macro and weighted F1-scores both at 0.98. The confusion matrix revealed that most misclassifications occurred between similar hand gestures, with only a few off-diagonal entries. Specifically, the model misclassified 1 instance of paper as rock and 2 instances of scissors as rock, indicating minimal confusion.

These results demonstrate that the deeper architecture of Model 3 provided improved generalization capabilities, offering a strong balance between capacity and regularization.

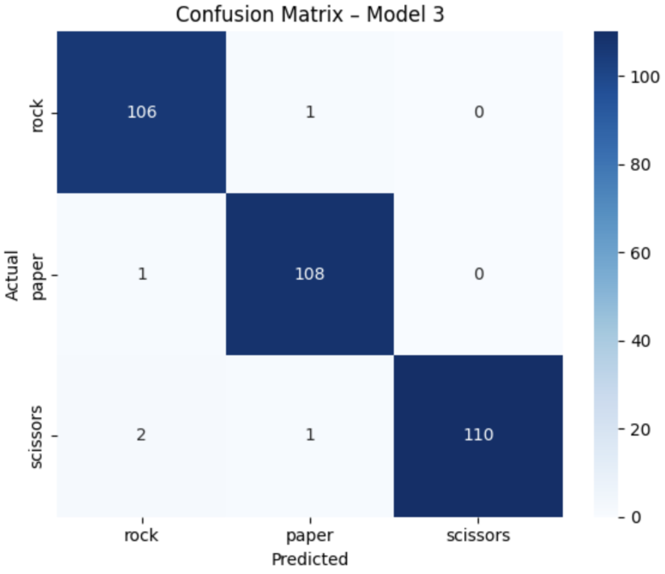

Figure 4: Confusion Matrix for Model 3

Model Comparisons

All three CNN architectures successfully classified rock-paper-scissors gestures with high accuracy. Model 3 achieved the highest performance, followed closely by Model 2. Despite its simplicity, Model 1 still performed remarkably well and demonstrated strong generalization. Given the low complexity of the task and clean dataset, even the simplest architecture was sufficient to produce reliable results. However, the deeper models offered marginal gains in precision and recall, particularly in minimizing misclassifications between similar gestures.

Conclusion

This project demonstrated the effectiveness of convolutional neural networks for image classification in a simple gesture recognition task. Starting from a basic architecture and gradually increasing complexity, three models were developed, trained, and evaluated using a well-balanced dataset. Proper data preprocessing, including normalization and augmentation, played a key role in enhancing model performance. Although deeper networks provided marginal improvements, the results suggest that for well-structured and low-variability tasks such as rock-paper-scissors classification, even relatively shallow CNNs can achieve high accuracy. This highlights the importance of matching model complexity to task difficulty in practical applications.

References

1. Kalra, K. (2021). Convolutional Neural Networks for Image Classification. *Medium*. Retrieved from `https://medium.com/@khwabkalra1/convolutional-neural-networks-for-image-classification-f0754f7b94aa`

2. Freeman, G. (2020). Rock-Paper-Scissors Dataset. *Kaggle*. Retrieved from `https://www.kaggle.com/datasets/drgfreeman/rockpaperscissors`

3. Analytics Vidhya. (2020). Image Augmentation on the Fly using Keras' ImageDataGenerator. Retrieved from `https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/`

4. Dutta, S. (2020). Designing Your Own Convolutional Neural Network (CNN) Model — A Step-by-Step Guide for Beginners. *Medium*. Retrieved from `https://medium.com/@sanjay_dutta/designing-your-own-convolutional-neural-network-c`