Hand-Drawn Doodle Classification: A Comparative Study of Machine Learning Approaches

Abstract

This project presents a comparative analysis of machine learning approaches for hand-drawn doodle classification using Google's QuickDraw dataset. We implemented and evaluated seven models spanning classical machine learning (Logistic Regression, Support Vector Machines with three kernels) and deep learning (three Convolutional Neural Network variants). Our experiments demonstrate that CNNs significantly outperform classical approaches, and that simplified CNN architectures maintain high accuracy while offering substantial computational advantages for deployment.

1. Dataset and Methodology

1.1 Dataset

We utilized Google's QuickDraw dataset, comprising 50 million hand-drawn images across 345 categories collected from over 15 million users worldwide. For computational tractability, we selected five diverse classes: **cat, tree, car, apple, and fish**, with 5,000 samples per class (25,000 total). These classes represent varying object types (organic vs manufactured), shape complexity (geometric vs irregular), and drawing consistency.

Preprocessing: Images are 28×28 grayscale, binarized at threshold 127 to reduce dimensionality while preserving essential structure. Data split follows standard practice: 70% training, 15% validation, and 15% test.

1.2 Models Implemented

Classical Machine Learning:

- **Logistic Regression:** Multinomial classifier, serving as baseline
- **SVM with Linear Kernel:** Maximum margin classifier
- **SVM with RBF Kernel:** Gaussian kernel for non-linear decision boundaries
- **SVM with Polynomial Kernel:** Degree-3 polynomial feature expansion

Deep Learning (Convolutional Neural Networks):

- **CNN v1 (Full):** Two convolutional layers (32 and 64 filters), two max-pooling layers, dense layer (128 units), dropout (0.25)
- **CNN v2 (Simplified):** Single convolutional layer (32 filters), max-pooling, dense layer (128 units), dropout
- **CNN v3 (Minimal):** Single convolutional layer (32 filters), max-pooling, dense layer (64 units), dropout

All CNNs use 3×3 filters, ReLU activation, Adam optimizer, and are trained with early stopping and learning rate reduction callbacks.

2. Experimental Results

2.1 Performance Comparison

Model	Test Accuracy	Training Time
Logistic Regression	86.7%	5.0s
SVM (Linear)	72.5%	19.4s
SVM (RBF)	90.7%	29.6s
SVM (Polynomial)	45.5%	34.5s
CNN v1	**95.5%**	62.9s
CNN v2	94.5%	33.2s
CNN v3	94.6%	27.0s

2.2 Key Findings

1. Deep Learning Superiority: CNNs achieve substantially higher accuracy than the best classical method.

2. Kernel Method Comparison: Within SVMs, performance ordering RBF > Polynomial > Linear directly correlates with kernel expressiveness.

3. Architecture Simplification: CNN simplification yields diminishing returns. Removing the second convolutional layer costs modest accuracy but provides significant training speed improvement. Further reducing the dense layer yields additional speedup with minimal accuracy loss.

4. Per-Class Performance: Car (geometric, consistent) and apple (simple shape) achieve highest accuracy across all models. Fish (high variability, similar to cat) proves most challenging. Common confusions (cat and fish, tree and apple) align with semantic and visual similarity.

5. Generalization: Small validation-test accuracy gaps across all models indicate robust generalization without overfitting, attributable to dropout regularization, early stopping, and dataset diversity.

3. Conclusions

CNNs substantially outperform the best classical method (SVM with RBF kernel), validating the importance of learned hierarchical features for image recognition tasks.

Critically, we demonstrate that CNN architectures can be substantially simplified for simple image data. Our minimal CNN achieves strong accuracy with significantly fewer parameters and faster training than the full model, making it optimal for resource-constrained deployment scenarios.