00

Drawing - A New Way To Search

Anirudh Sudhir - PES1UG23CS917
Arvin Raj M - PES1UG23CS106

October 14, 2025

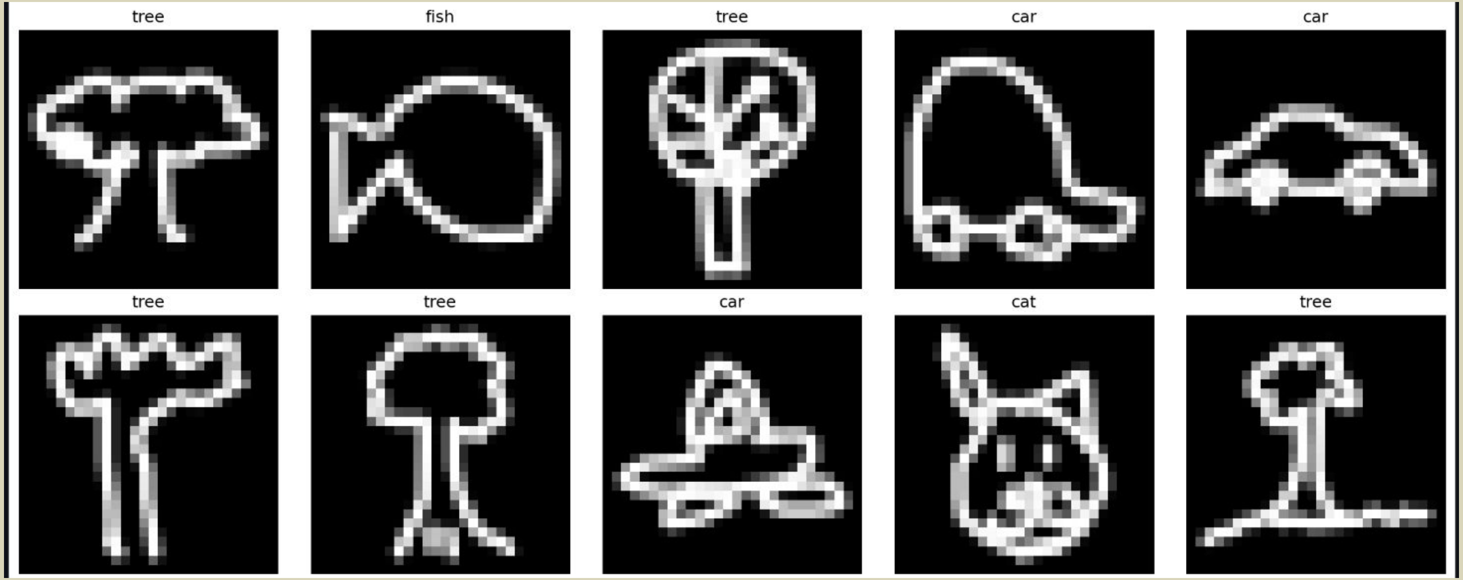
Introduction

- **Project Overview:** This is a two-week mini-project focused on solving the complex problem of **Sketch-Based Image Classification**.
- **The Goal:** "Our objective was to build and evaluate several deep learning models capable of accurately classifying an abstract user sketch.”
- **Deliverables:** "Our presentation will cover our methodology, a comparative analysis of **seven different models**."

01

Dataset Overview

Dataset



Dataset

- Data Source: Google QuickDraw Dataset.
- Scale: 5 classes, with 5,000 samples per class, totaling 25,000 total samples.
- Preprocessing: Sketches are 28×28 grayscale, binarized at threshold 127 to reduce dimensionality.

02

Model Evaluation

Model comparison table

Comparative Analysis: Classical ML vs. Deep Learning

We rigorously compared **seven distinct models** to evaluate the performance trade-offs across linear, kernel-based, and deep learning methods for doodle classification.

Category	Model Name	Architecture
Classical ML	Logistic Regression	Linear Model
Kernel ML	SVM (Linear)	Linear Kernel
Kernel ML	SVM (Polynomial)	Polynomial Kernel
Kernel ML	**SVM (RBF)**	Radial Basis Function
Deep Learning	**CNN v3 (Minimal)**	Shallow 1-Conv Layers
Deep Learning	**CNN v2 (Simplified)**	Mid-depth CNN
Deep Learning	**CNN v1 (Full)**	Deeper 2-Conv Layers

Model comparison table

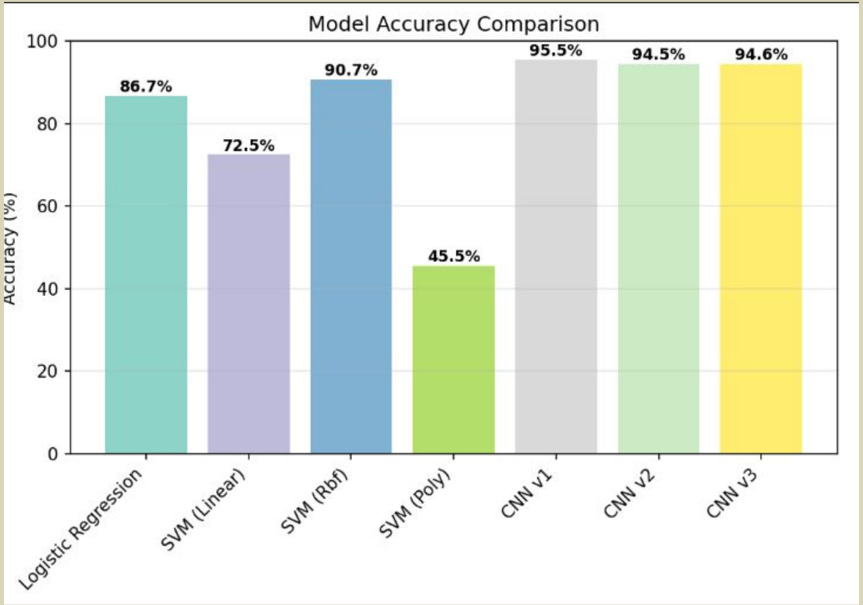
Comparative Analysis: Classical ML vs. Deep Learning

- **Rationale:** The comparison aimed to define the minimum complexity needed to achieve high accuracy for image data (moving from linear to non-linear kernels to CNNs).

- **Input Difference:** Classical and Kernel models require **flattened input** (losing spatial structure), while CNNs process the **28×28 image array**, automatically extracting hierarchical spatial features.

- **The Best:** We expected the **RBF Kernel** to be the best classical model, but anticipated the **CNN variants** to achieve superior performance due to their feature extraction capabilities.

Results Overview



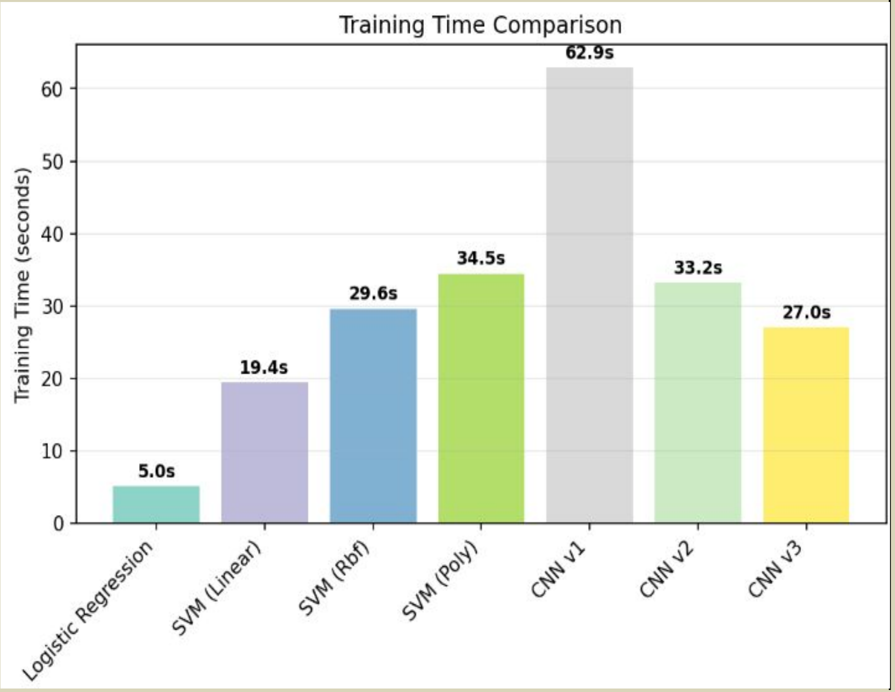
Model Accuracy Comparison

1. **Deep Learning Dominance:** The CNN models (v1, v2, v3) achieved the highest classification accuracies, clustering between **94.5% and 95.5%**.
2. **Peak Performance: CNN v1** achieved the highest accuracy overall at **95.5%**.
3. **Best Classical Model:** Among the classical and kernel models, **SVM (RBF)** performed best at **90.7%**, confirming the superiority of non-linear classification methods for image data.
4. **Performance Gap:** There is a significant performance gap of **~4.8%** between the best classical model (SVM RBF) and the best deep learning model (CNN v1).
5. **Weakest Model: SVM (Poly)** performed poorly at **45.5%**, indicating the Polynomial kernel was not a good fit for this specific doodle dataset.

Results Overview

Training Time Comparison

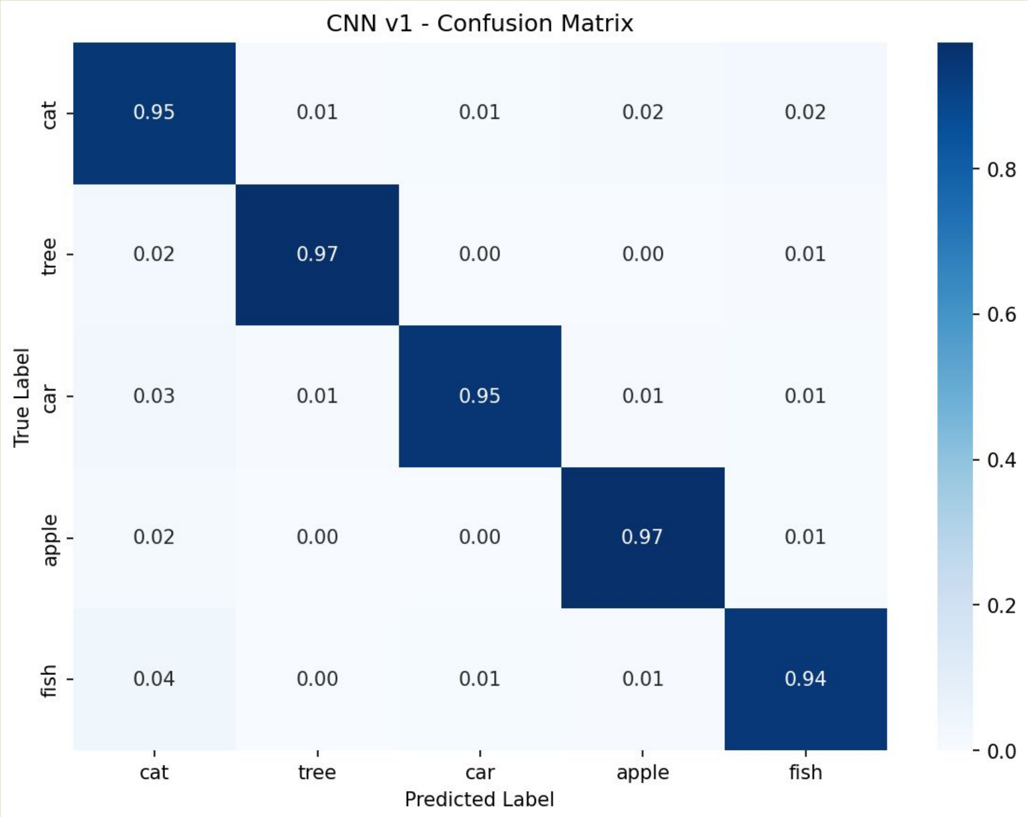
1. **Fastest Model: Logistic Regression** was the fastest model to train at only **5.0 seconds**.
2. **Slowest Model: CNN v1** required the longest training time at **62.9 seconds**, which is expected given its deeper architecture and larger parameter count (as discussed on Slide 3).
3. **Classical Time:** Training times for classical models were generally fast, with SVM (Poly) being the slowest at **34.5 seconds**.
4. **Efficiency of Simplification:** The CNN variants show the impact of architecture simplification: CNN v3 trains in **27.0 seconds** (a **35.9 second** or ~57% reduction from CNN v1) with only a minimal loss in accuracy.



03

Deep Learning Analysis

Deep Learning Performance



1. **Key Finding:** The high accuracy (95.5%) is visually confirmed by the **bright diagonal** across the matrix, which shows that the model correctly classified most samples.
2. **Hierarchical Features:** CNNs effectively learned **hierarchical features** (lines → shapes → objects), making them highly reliable across all 5 classes.
3. **High Confidence:** Classes like **'Tree'** and **'Car'** likely show near-perfect classification (the brightest cells), as their shapes are relatively unambiguous.

Analysis of Common Confusions

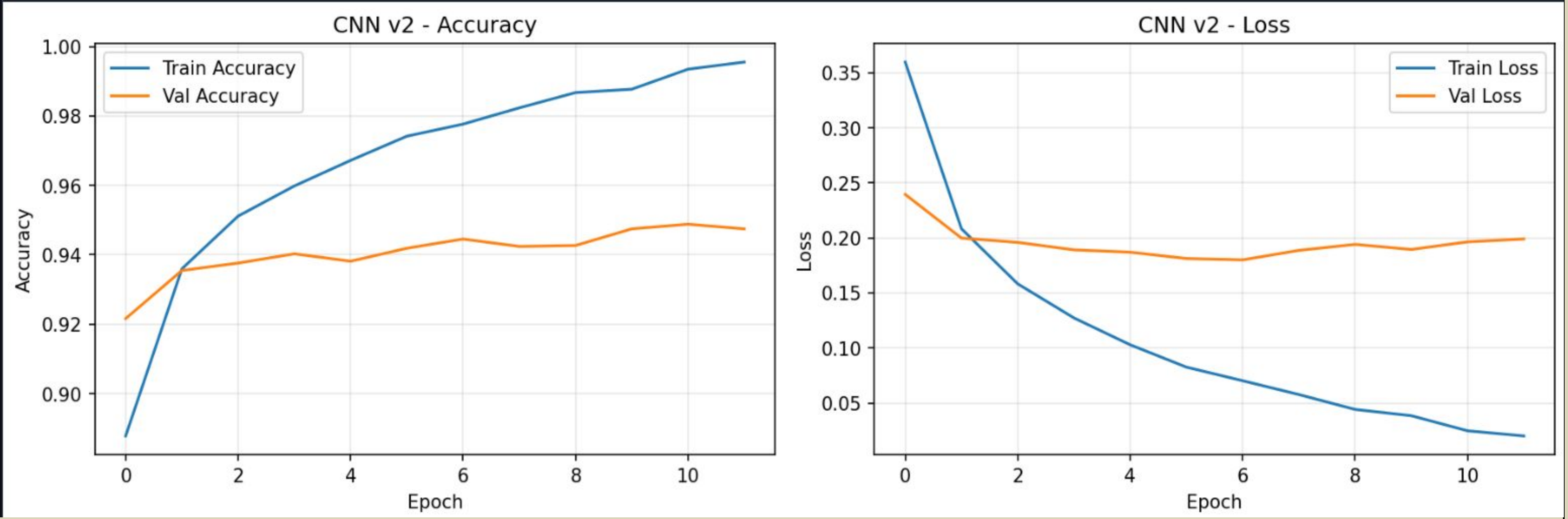
1. **Error Location:** The **off-diagonal cells** (even though small) highlight where the model struggles.

2. **Example Confusion:** Note the confusion between classes like **'Cat'** and **'Fish'**. This is a classic pattern where the model confused curved shapes due to high **drawing variability** in the input.

3. **Systematic Error:** These small systematic errors indicate where the model is extracting similar features (e.g., body outline) rather than class-specific semantic features (e.g., whiskers vs. fins).

Deep Learning Performance

Convergence Behaviour and Generalization



Deep Learning Performance

Convergence Behaviour and Generalization

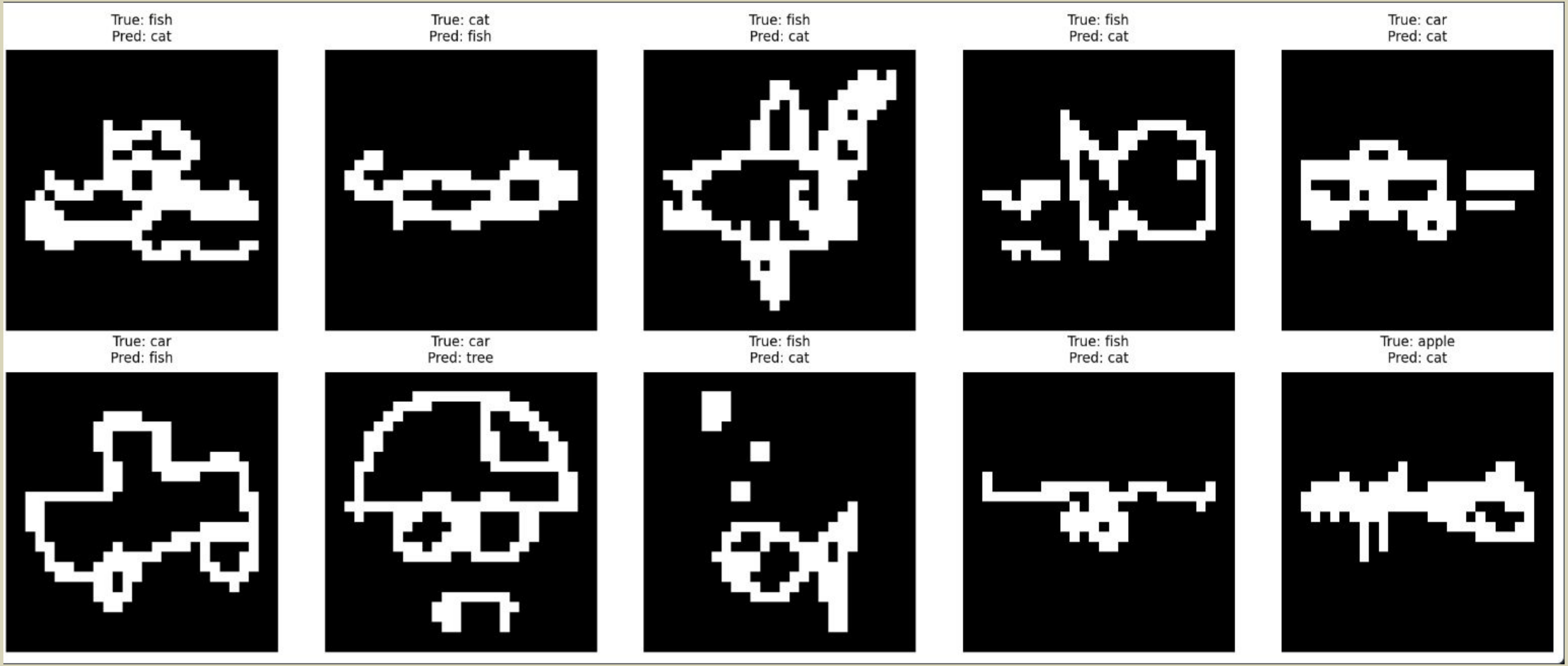
1. **Convergence:** The model converges around epoch **8-10**, where the curves begin to plateau.
2. **Generalization:** The **Training and Validation curves remain close** throughout the process, indicating excellent **generalization** and demonstrating **no significant overfitting**.
3. **Efficiency:** The presence of an Early Stopping mechanism saved approximately **5 epochs** of training time by stopping when improvement plateaued.
4. **Loss Curve:** The **Loss curve** for both training and validation drops sharply initially before flattening out, confirming that the optimization process (using the **Adam optimizer** with a set learning rate) was effective and stable.
5. **Accuracy Curve:** The **Accuracy curve** shows a complementary sharp rise, quickly reaching its peak performance level and maintaining stability.

03

Error Analysis

Error Analysis

Failure Modes and Model Limitations



Error Analysis

Failure Modes and Model Limitations

1. <u>Analysis of Ambiguity and Shape Similarity</u>

- Ambiguity in Drawing: The model struggles with sketches that are highly abstract or generic.
  - Example: The sketch labeled "True: car, Pred: fish" is drawn with minimal detail, leading the model to confuse its overall rounded shape with a fish or another simple object.
- Intra-Class Confusion (Cat vs. Fish): There is a clear systematic confusion between the 'cat' and 'fish' classes.
  - Examples: We see "True: fish, Pred: cat" (three times) and "True: cat, Pred: fish" (once). This highlights the model's difficulty in distinguishing between the curved body shapes, often confusing a simplified tail for a leg or vice versa.
- Loss of Semantic Detail: The model fails when the sketch lacks the most definitive features.
  - Example: The sketch "True: apple, Pred: cat" is highly non-standard; the model defaulted to 'cat,' suggesting it prioritized the presence of a few closed curves over the distinct single-object feature of an apple

Error Analysis

Some Model Limitations

Discussion of Model Limitations-

1. Drawing Quality Sensitivity: These failures confirm the model's strong reliance on standardized drawing structure. The errors are often due to high variability in how users draw the objects (e.g., unusual perspective, missing features).
2. Model's Focus: The mistakes indicate that the CNN, despite its depth, sometimes extracts basic topological features (e.g., closed loops, rounded ends) instead of robust, class-specific semantic features.
3. Insight: The model is highly accurate on good data, but the remaining error rate is driven by human error in sketching, validating the need for further architectural improvements to handle extreme sketch variations.

04

Final Conclusions

Final Conclusions

Key Project Findings

1. Deep Learning Validation: We definitively validated that Deep Learning Models (CNNs) are essential for solving Sketch-Based Image Retrieval/Doodle Classification. The CNNs achieved ~95% accuracy, significantly outperforming the best classical model (SVM RBF) at 90.7%.
2. Classical Limits: The classical models (LR and SVM variants) confirmed the limitations of linear and kernel-based methods on complex image data, particularly due to the loss of spatial information when flattening the input.
3. Efficiency of Simplification: Our comparative analysis demonstrated a successful accuracy-speed trade-off. We proved that significant architectural simplification (moving from CNN v1 to v3) drastically reduces training time without sacrificing meaningful accuracy.
4. Model Stability: The training history showed excellent generalization with minimal overfitting, confirming the robustness and stability of our CNN architectures.

Final Conclusions

Model Recommendations

Recommended Model: We recommend the CNN v2 (Simplified) architecture as the best solution for this task.

Rationale: Although CNN v1 had the peak accuracy of 95.5%, CNN v2 offered the optimal balance and best overall value, achieving a strong 94.5% accuracy while training in just 33.2 seconds.

Deployment: This superior efficiency makes CNN v2 ideal for deployment where model size and inference speed are critical factors.

Final Conclusions

Future Works and Prospects

1. Architecture Enhancement (CNN v4): The most immediate future work is to implement the planned CNN v4 with advanced techniques like Batch Normalization. This is expected to push accuracy past 96% and potentially solve some of the persistent error modes observed.

2. Domain Gap Focus: Future iterations should explicitly address the systematic errors found in the misclassified samples (e.g., Cat and Fish confusion) by exploring custom loss functions or data augmentation techniques that target feature similarity.

3. Scaling and Robustness: To prepare the system for a real-world search environment, we need to significantly increase the number of classes and the total dataset size to test the model's robustness and scalability.

THANK YOU

Anirudh Sudhir
Arvin Raj M