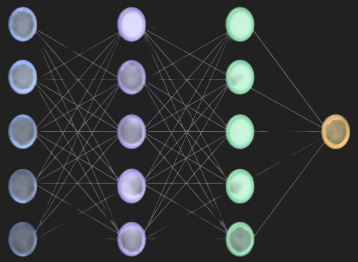


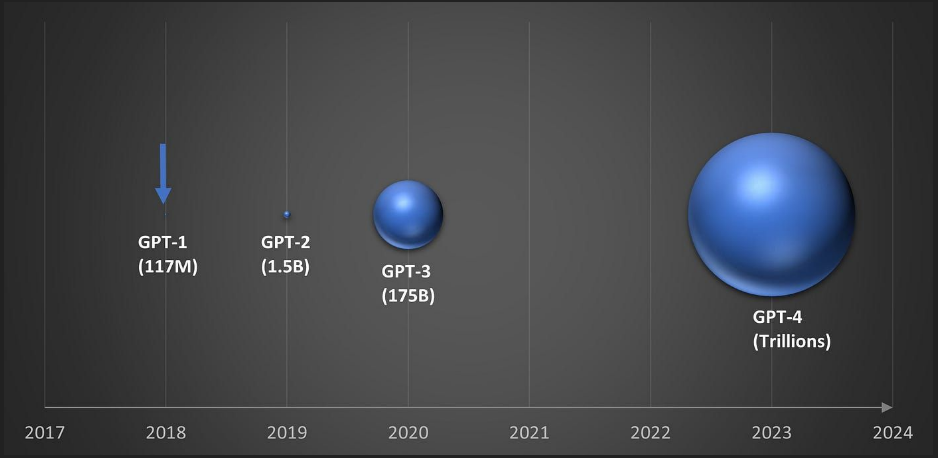
Can a Custom CNN Compete with OpenAI's GPT-4o?

Sushmit Chakma, Alan Liu

Motivation

The Rise of Giant AI Models:

- OpenAI's GPT-4o: **1.8 trillion parameters**
- Trained on billions of images from the internet
- Zero-shot capability: No training needed for new tasks



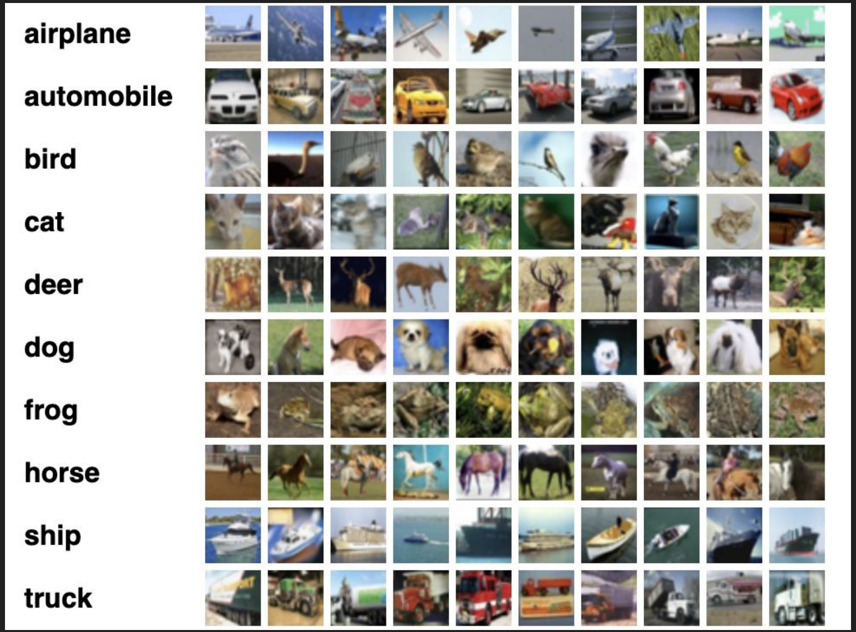
Our Question

How does a ***trained custom CNN*** perform on an ***image-classification dataset*** compared to ***GPT-4o Vision, a general-purpose model*** that is ***not fine-tuned on the dataset*** and performs zero-shot classification?

Image Source: Eversberg, L. (2024, February 24). Size Matters: How Big Is Too Big for An LLM? Towardsai.net. https://towardsai.net/p/artificial-intelligence/size-matters-how-big-is-too-big-for-an-llm#:~:text=As%20shown%20in%20the%20graph%20above%2C%20GPT%2D1,size%20for%20each%20new%20iteration%20of%20GPT.

CIFAR-10 Dataset

- Total images: **60,000**
- Training set: **50,000**
- Test set: **10,000**
- Classes: **10**
- Image size: **32×32 pixels (tiny!)**



Sources: CIFAR-10 and CIFAR-100 datasets. (2025). Toronto.edu. https://www.cs.toronto.edu/~kriz/cifar.html
Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

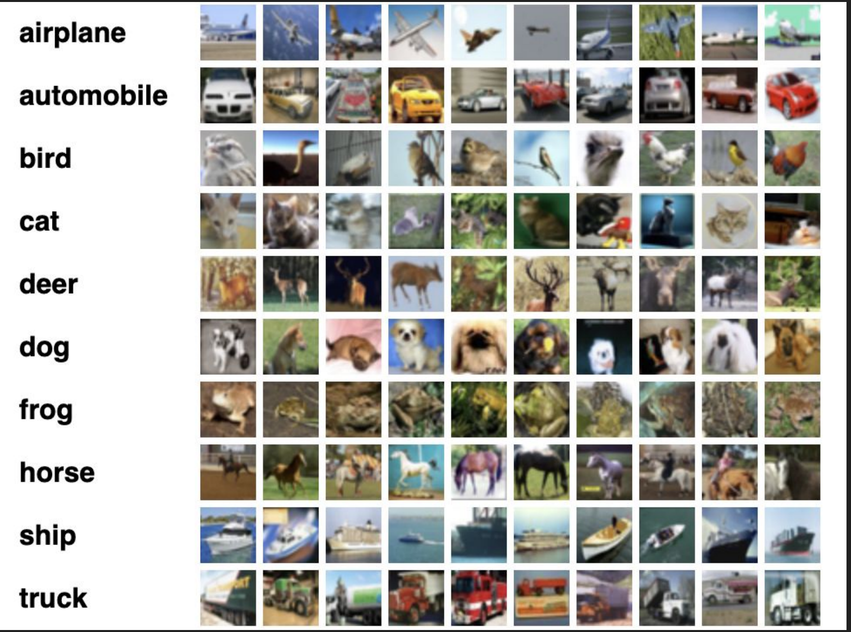
Testing Size

- 2,000 images

  (from 10,000 total to save API costs)
- **Stratified sampling:**

  Exactly 200 random images per class
- **Saved indices:**

  stratified_subset_2000.json
- Both CNN & GPT-4o tested on identical images



Sources: CIFAR-10 and CIFAR-100 datasets. (2025). Toronto.edu. https://www.cs.toronto.edu/~kriz/cifar.html
Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

CNN with PyTorch

2-layer CNN

- Convolutional (extract features), pooling (downsample features) layers
- Conv → Pool → Conv → Pool → Fully Connected Model
- 200 epochs, or stopping after 20 w/no change (overfitting)
- Data augmentation: randomly flip/crop images in training
- NVIDIA CUDA Parallel Computing

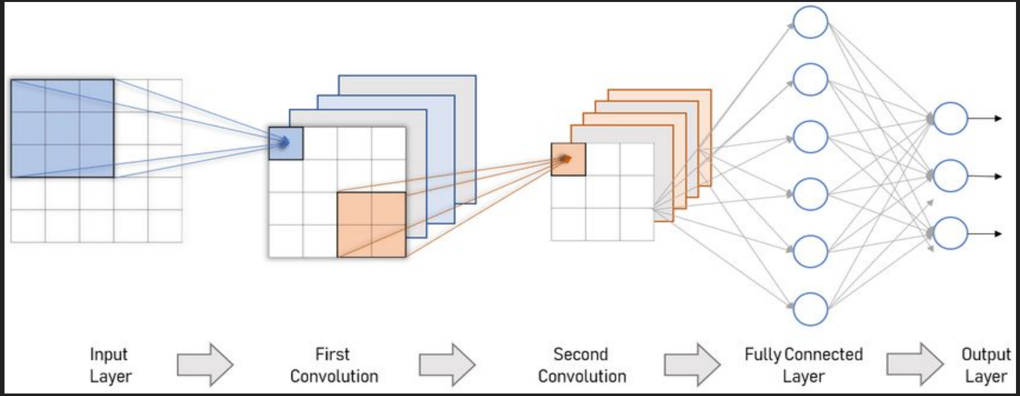
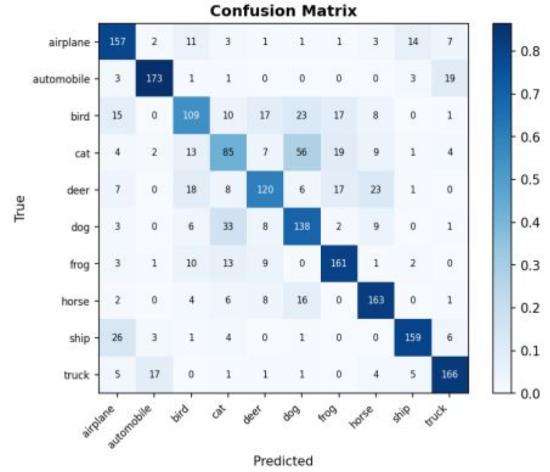
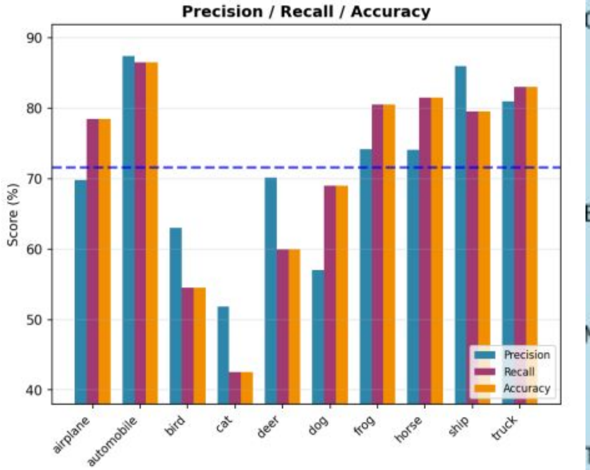


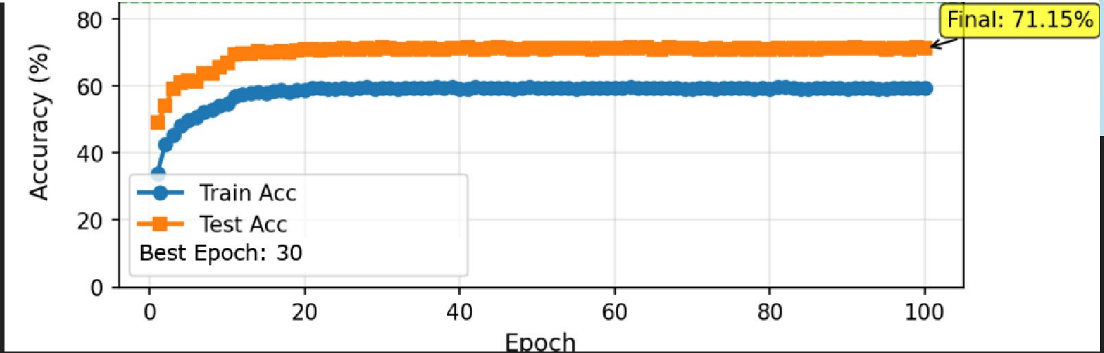
Image source: A Deep Learning Based Framework for Physical Assets' Health Prognostics Under Uncertainty for Big Machinery Data - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Two-layer-CNN-example_fig3_342365956

CNN Results



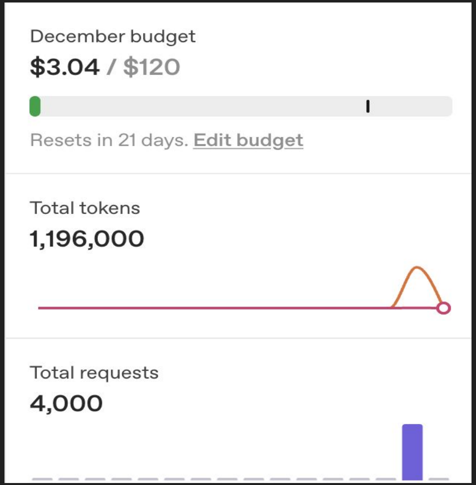



```
Overall Metrics:
  Accuracy:         71.55%
  Correct:          1431 / 2000

  Macro Precision: 71.44%
  Macro Recall:    71.55%
  Macro F1:        71.25%

Best Performing Classes:
  automobile:     86.5%
  truck: 83.0%
  horse:    81.5%

Most Challenging Classes:
  deer:        60.0%
  bird:        54.5%
  cat:     42.5%

Top Confusion Pairs:
  cat -> dog: 56 errors
  dog -> cat: 33 errors
  ship -> airplane: 26 errors
  bird -> dog: 23 errors
  deer -> horse: 23 errors
```



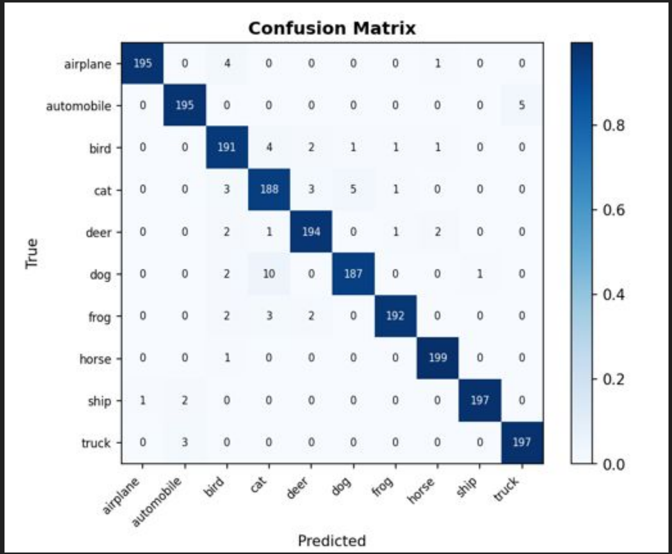
GPT-4o API

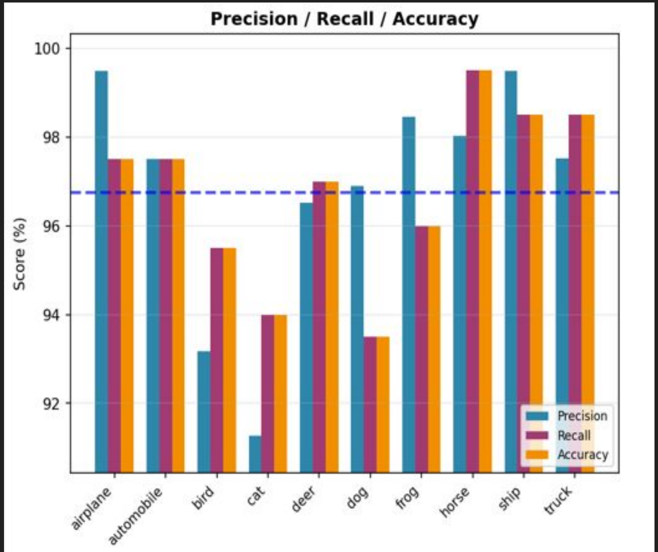
- CIFAR image (raw tensor) ➔ PNG ➔ Base64 ➔ GPT-4o API ➔ Prediction
- Each API call cost : $0.00076 & 299 tokens
- We ran 4000 API calls




```
{
    "type": "text",
    "text": "Classify this image as exactly one of: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. Return only the label, nothing else."
},
{
    "type": "image_url",
    "image_url": {
        "url": f"data:image/png;base64,{img_base64}"
    }
}
```

GPT-4o Results





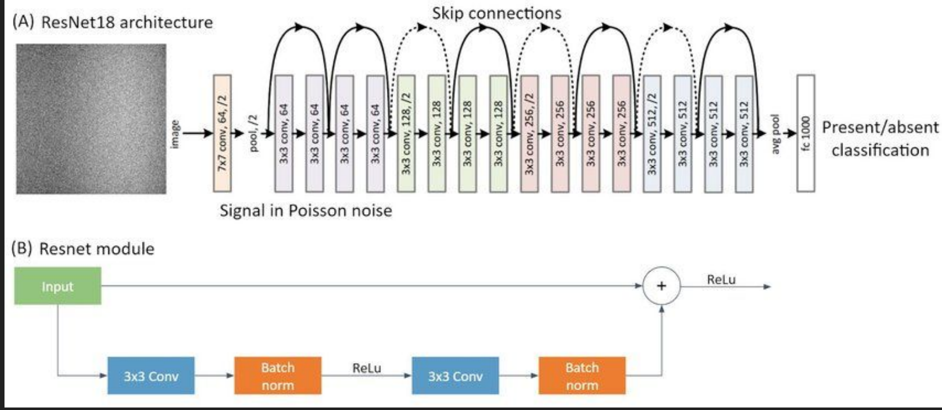

```
Overall Metrics:
  Accuracy:         96.75%
  Correct:          1935 / 2000

  Macro Precision: 96.83%
  Macro Recall:    96.75%
  Macro F1:        96.78%

Best Performing Classes:
  ship:      98.5%
  airplane: 97.5%
  horse:     99.5%

Most Challenging Classes:
  cat:       94.0%
  dog:       93.5%
  frog:      96.0%

Top Confusion Pairs:
  dog -> cat: 10 errors
  automobile -> truck: 5 errors
  cat -> dog: 5 errors
  airplane -> bird: 4 errors
  bird -> cat: 4 errors
```

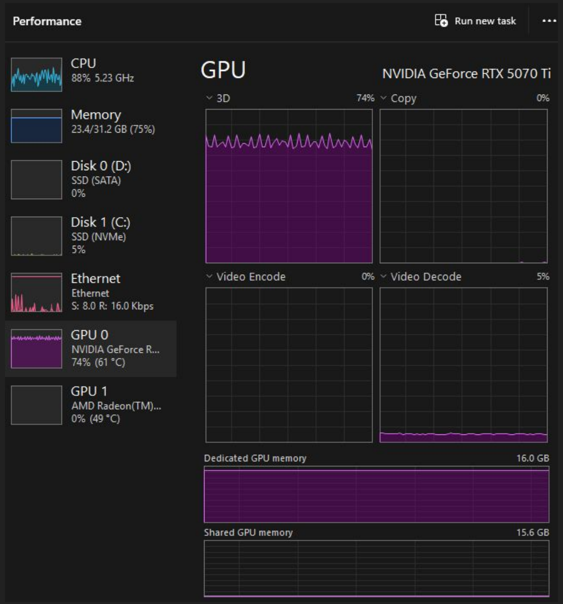
Model	Accuracy
Custom CNN	71.55%
GPT-4o	96.75%

Gap: 25.2% !!!

CNN Improvements

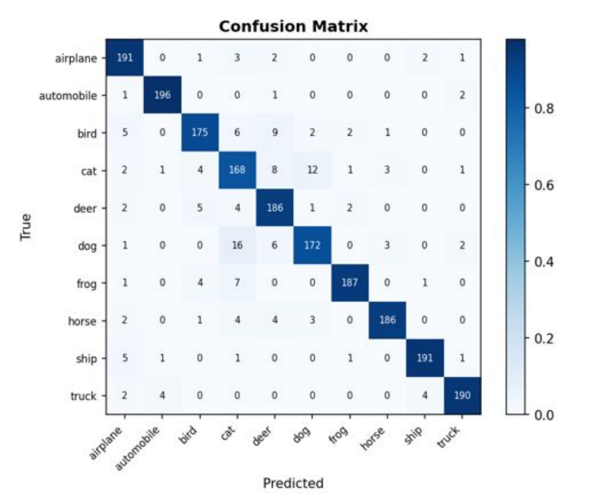
- Resnet style architecture
  - Increased depth: 2 conv layers → 7 conv layers (6 residual blocks)
  - Skip connections (direct gradient flow)
- More channels
  - More filters & more complex patterns
  - More GPU usage! (due to CUDA)
- Better data augmentation (generalization)
  - Keep random crop and flip
  - Random color jitter and rotation
  - Randomly erase patches
- Sophisticated training techniques
  - Label smoothing, adaptive learning rates

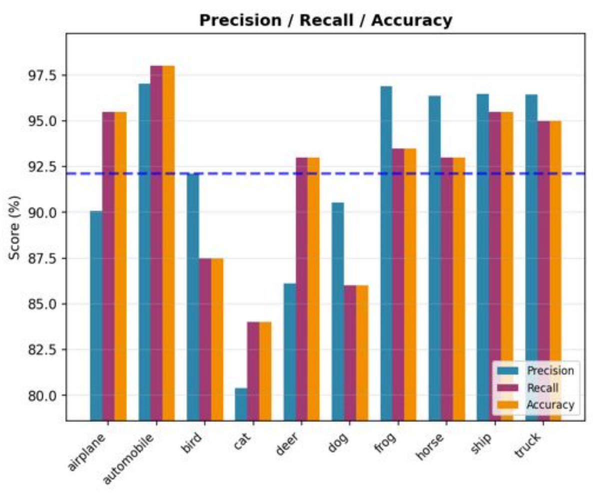




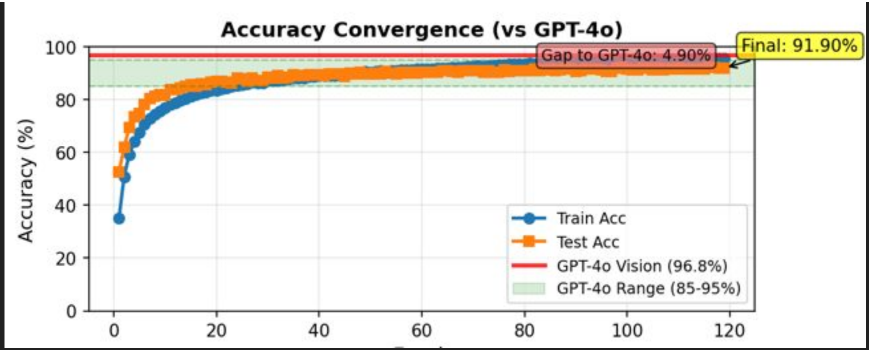
Source: https://www.researchgate.net/figure/The-ResNet-convolutional-neural-network-architecture-A-The-input-signal-plus_fig1_337208775

Improved CNN Results






```
Overall Metrics:
  Accuracy:         92.10%
  Correct:          1842 / 2000

  Macro Precision:  92.24%
  Macro Recall:     92.10%
  Macro F1:         92.13%

Best Performing Classes:
  automobile:      98.0%
  airplane: 95.5%
  ship:     95.5%

Most Challenging Classes:
  bird:      87.5%
  dog:       86.0%
  cat:      84.0%

Top Confusion Pairs:
  dog -> cat: 16 errors
  cat -> dog: 12 errors
  bird -> deer: 9 errors
  cat -> deer: 8 errors
  frog -> cat: 7 errors
```



Results

Model	% Accuracy
Baseline CNN accuracy	71.55%
Improved CNN accuracy	**92.10%** (+20.55%)
GPT-4o accuracy	96.75%

Most Error-Prone:

Model	Cat → Dog	Dog → Cat
Baseline	56	33
Improved	12	16
GPT-4o	5	6

Key Takeaways

- LLMs still have top accuracy
  - Massive pre-training
- ResNet increases CNN accuracy considerably
  - Architecture matters
- Heavily tuned CNNs have competitive accuracy rates

Future Work

- Compare it with open-weight LLMs (i.e. LLaVA)
- Train and test on complex datasets (i.e. ImageNet)

THANK YOU!

Any questions?