LEAN VS MESSY ROOM DETECTOR

Project Tier - 1

Somayeh Choboderazi

Professor Patricia McManus

Computer Vision - Artificial Intelligence 1378

October28, 2025

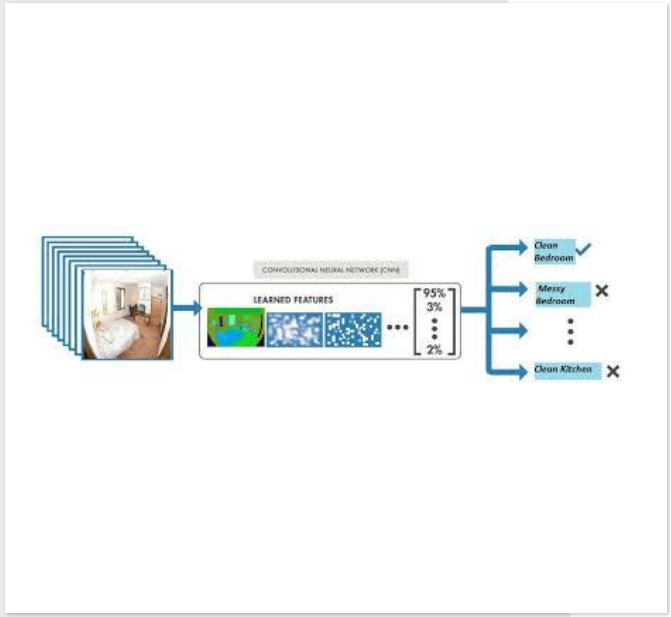
THE PROBLEM

Problem: Many people spend extra time cleaning and organizing rooms.

Who cares: Cleaning companies, homeowners, students, and hotels.

Why it's important: A clean space saves time, improves mood, and boosts productivity.

- People spend time checking and cleaning rooms manually
- Hard to know when a room needs cleaning
- Affects homeowners, hotels, offices, and cleaning services
- Wastes time and labor costs daily
- Manual inspection is inconsistent and tiring
- Need for an automatic cleanliness detection system



SOLUTION (OVERVIEW)

Solution: AI system that classifies room photos as *clean* or *messy*.

How it works:

- AI model analyzes a photo of a room
- Predicts “clean” or “messy” instantly
- Uses image classification to learn visual patterns
- Helps people and companies manage cleaning better
- Practical, real-world tool — not just theory
- Can extend to smart homes or cleaning robots later

TECHNICAL APPROACH

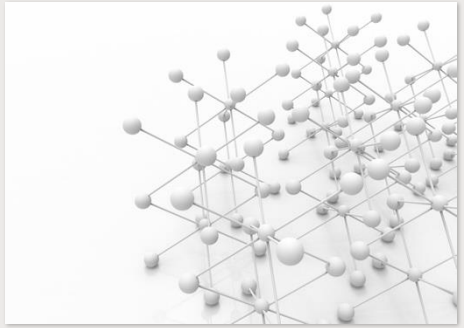
Technique: Image Classification

Model: ResNet50 (pre-trained)

Framework: PyTorch in Google Colab

Why: ResNet50 is accurate, light, and great for small datasets.

- Model: ResNet50 pre-trained CNN
- Framework: PyTorch on Google Colab
- Transfer learning for faster results
- Learns patterns like clutter, shape, and texture
- Uses cross-entropy loss + Adam optimizer
- Decision boundary separates “clean” vs “messy” images



DATA PLAN

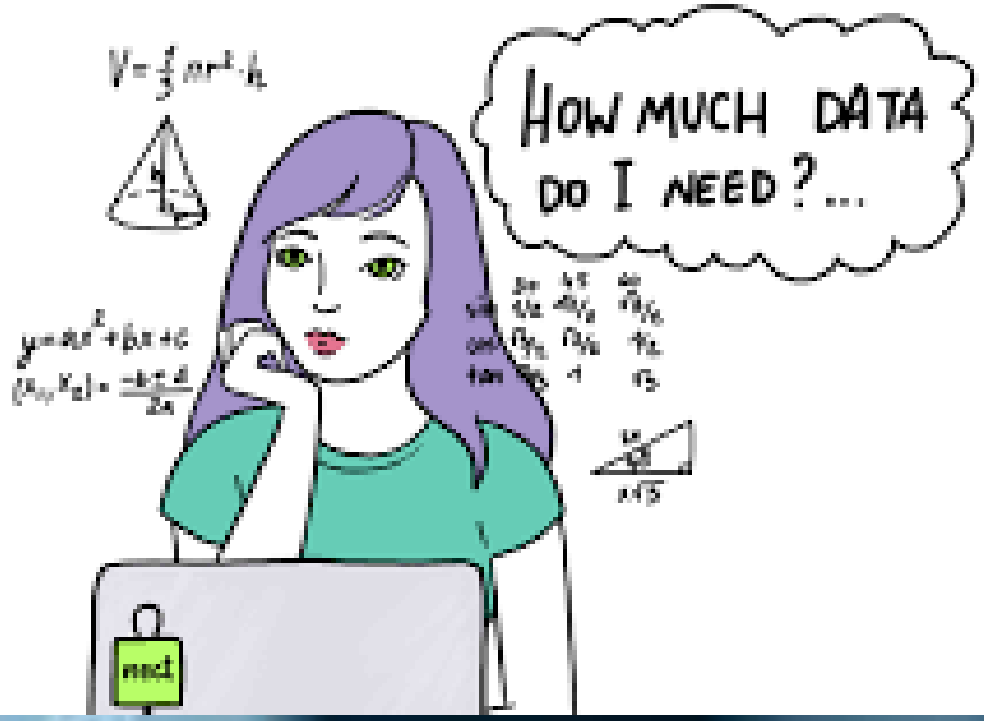
Data Source: Google Images or own photos

Classes: “Clean” and “Messy”

Size: 100–200 images

Preparation: Resize, label, and clean dataset.

- Source: Google Images + my own photos
- Two classes: “Clean” and “Messy”
- Dataset size: around 200 images
- Split 70/20/10 for train, validation, test
- Preprocess: resize, normalize, and label
- Augment data to handle lighting and angle differences



SYSTEM DIAGRAM

[Input Image]

↓

[ResNet50 Model: Feature Extraction + Classification]

↓

[Output: "Clean" or "Messy”

- Input: Room photo uploaded or captured
- Preprocessing: resize and normalize image
- Feature extraction: ResNet50 backbone
- Classification: fully connected layer
- Output: “Clean” or “Messy” prediction
- Visual feedback shown to the user

SUCCESS METRICS

Metric	Example	Target
Accuracy	Classification accuracy	≥ 85%
Speed	Inference time	< 1 sec
Data	~200 images	—

- Target accuracy: ≥ 85%
- Inference speed: < 1 second per image
- Reliable predictions on new data
- Good balance between accuracy and speed
- Test with unseen rooms for validation
- Measure confusion matrix to find weak areas

WEEK-BY-WEEK PLAN

- Week 10: Collect and label dataset
- Week 11: Train and tune model
- Week 12: Test and improve accuracy
- Week 13: Create demo and visuals
- Week 14: Final testing and documentation
- Week 15: Presentation and submission

Week	Task	Milestone
10 (Oct 30)	Collect dataset	Dataset ready
11 (Nov 6)	Train model	Model working
12 (Nov 13)	Test & improve	Accuracy ≥85%
13 (Nov 20)	Create demo	Demo ready
14 (Nov 27)	Final docs	Report done
15 (Dec 4)	Present	🎉

CHALLENGES & BACKUP PLANS

Challenges:

- Not enough data → add more from Google
- Low accuracy → use stronger model or augmentation
- Lighting or angles cause confusion

Backup Plans:

- Colab runtime limits training time
- Overfitting → use dropout or early stopping
- Backup: pretrained models + smaller batch sizes

RESOURCES NEEDED

Resource	Description
Compute	Google Colab (Free GPU)
Framework	PyTorch, Torchvision
Dataset	100–200 room images
Cost	$0
Tools	Colab, PowerPoint, GitHub

- Hardware: Google Colab GPU
- Software: PyTorch, OpenCV, TorchVision
- Dataset: public + personal images
- Version control: GitHub repository
- Tools: PowerPoint for slides, Markdown for docs
- Cost: $0 (all open-source tools)

SUMMARY

- Built an AI system to classify rooms as clean or messy
- Uses pre-trained ResNet50 with transfer learning
- Achievable within limited data and resources
- Solves a real-world problem efficiently
- Can evolve into smart home or facility management tools
- Simple idea, strong practical impact

REFERENCES

•He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition (ResNet)*. CVPR.

•PyTorch. (2025). *PyTorch Documentation.* Retrieved from https://pytorch.org

•Roboflow. (2024). *Open Image Datasets for Computer Vision.* Retrieved from https://roboflow.com/datasets

•Shorten, C., & Khoshgoftaar, T. M. (2019). *A survey on Image Data Augmentation for Deep Learning.* Journal of Big Data.

•scikit-learn. (2025). *Metrics and Model Evaluation.* Retrieved from https://scikit-learn.org

•OpenCV. (2025). *Computer Vision Library.* Retrieved from https://opencv.org

•Google Colab. (2025). *Colab Documentation.* Retrieved from https://colab.research.google.com