AI & Image Recognition for Mobile Apps

Andrew Burks
Director, Decision Sciences Visual Analytics, Epsilon
PhD Candidate, Electronic Visualization Laboratory, UIC

Epsilon | PUBLICIS GROUPE

evl electronic visualization laboratory

UIC

Roadmap

Foundations
What is machine learning? How do machines see images? What kinds of learning exist?

Building
Training a model from scratch vs. building on top of existing work. What happens during training vs. when the model is actually used?

What Models Can Do
The different jobs image models perform: creating, labeling, cutting out, and locating.

Shipping & The Future
How big are these models? What fits on a phone? How does image recognition actually get into a mobile app? And where is all of this headed?

Foundations

Introduction to Machine Learning

Developing an Eye for Design

- Internalized ***"taste"*** for good design develops gradually by absorbing thousands of examples, not from a checklist of what makes certain designs good vs. bad

- Machine learning works in the same way: rather than giving the system explicit rules, it develops an internal sense of concepts based on examples provided

- **Traditional programming** is writing rules by hand, **Machine learning** is showing enough examples for rules to emerge on their own

- ***These emergent rules are abstract and difficult to explain***

➔ Goodfellow et al., "Deep Learning" (MIT Press, 2016)
➔ cs231n.stanford.edu

How Neural Networks See Images 1/2

Breaking Down a Composition

- A design contains a typographic hierarchy, color relationships, a grid, negative space, and other Gestalt principles

- Neural networks learn to see images from its components in a similar way, in layers of increasing complexity

- This layered approach is called a Convolutional Neural Network (CNN)

➔ Zeiler & Fergus, "Visualizing CNNs" (2013)
➔ Olah et al., "Feature Visualization," distill.pub (2017)
➔ 3Blue1Brown neural network series (YouTube)

How Neural Networks See Images 2/2

Breaking Down a Composition

	Image Model	Design
Layer 1 **Pixels**	Raw numbers: (R, G, B, A)	Individual paint strokes without context
Layer 2 **Edges & Gradients**	Lines and boundaries	The grid in a layout
Layer 3 **Textures & Patterns**	Repeated structures, fur, fabric, brick	A visual motif
Layer 4 **Parts & Objects**	Eyes, wheels, leaves	A component of a design system
Final Output	"It's a cat"	The full composition, understood

- ➔ Zeiler & Fergus, "Visualizing CNNs" (2013)
- ➔ Olah et al., "Feature Visualization," distill.pub (2017)
- ➔ 3Blue1Brown neural network series (YouTube)

Supervised vs. Unsupervised Learning

Art school vs. a museum visit

Supervised Learning is like learning from a hands-on professor. Every example comes with a clear label and feedback. These machine learning models learn from labeled data: every image is labeled with the correct answer.

Unsupervised Learning is like wandering a museum alone with no plaques to guide you. Over time, you start grouping things yourself and seeing similarities and shared qualities. These machine learning models find hidden patterns and groups without labeled data.

Supervised learning is the standard approach for most image recognition that you would do in a mobile application.

- ➔ Bishop, "Pattern Recognition & Machine Learning" (2006)
- ➔ fast.ai practical deep learning course

How Models Learn: Optimization 1/2

Iterating through critique

You start with a "prior belief" of what is a good design solution for a project. You create your draft and present it for a **Critique Cycle**:

1. Present a draft
2. Get a critique
3. Identify what to fix
4. Revise and resubmit

Based on the feedback - what to change and how drastically to change it - you will **update** your design, and also your intuition of how to solve this problem in the future.

This is a statistical process called Bayesian reasoning: **guess, get feedback, revise**

How Models Learn: Optimization 2/2

Iterating through critique

An **image model** learns through the exact same cycle:

1. Feed an image to the model to get a prediction
2. Compare the prediction to the correct answer and get a score for the prediction
3. Figure out which internal setting caused the mistake
4. Adjust the setting slightly in the correct direction

Repeated **millions** of times, the model's predictions become consistently close to correct

➔ Ruder, "An Overview of Gradient Descent Optimization" (2016)
➔ 3Blue1Brown, "Neural Networks" series (YouTube)

Building

Training a Model from Scratch 1/2

Building a brand from nothing

To build the brand for new startup from a blank canvas, you would need to:

- Research your market
- Hire a creative team
- Invest in tools and studio space
- Spend months of iteration to develop a cohesive visual language

Training a Model from Scratch 2/2

Building a brand from nothing

Building an image model on your own:

- **Data:** Simple tasks (2-5 categories) needs 1k-10k images, Complex tasks (1k+ categories) need millions of images. ImageNet has 14M images from 21k categories
- **Time:** ***Small models*** = 1h - 1d, ***State-of-the-art models*** = weeks on a cluster, ***Foundation models*** = months on a massive cluster
- **Team: Small** = 1-2 ML Engineers, **Production** = 3-8 people, **Large-scale** = 20-100+
- **Cost:** $100 to $10M+
- **Tools:** Python with PyTorch, TensorFlow/Keras, Jax, and HuggingFace
- **Open Datasets:** ImageNet (14M), OpenImages (9M), COCO (330k)

➔ pytorch.org
➔ tensorflow.org
➔ image-net.org
➔ cocodataset.org

Transfer Learning & Fine-Tuning

Building on an existing design system

Alternatively, start with an established design system (Material Design, Apple's HIG) and its typography, color palettes, spacing, and components. Only make smaller customizations for your specific project

Transfer Learning starts with an existing model and fine-tunes it for your task.

- **Data:** 100-1k images
- **Time:** minutes to hours
- **Accuracy:** Often 90%+ for your task
- **Cost:** $0-$50
- **Tools**: Hugging Face , Google Teachable Machine , Apple Create ML

➔ Yosinski et al., "How transferable are features in deep neural networks?" (2014)
➔ huggingface.co
➔ teachablemachine.withgoogle.com
➔ developer.apple.com/create-ml

Training vs. Inference

Developing a brand identity vs. applying it

Developing a brand identity takes weeks-to-months of research and is expensive

Applying the brand identity to a deliverable is very fast to execute

Machine learning has the same two phases:

Training *(developing the brand)* happens "offline" with massive compute, using the entire dataset, to learn, evaluate, and adjust the model

Inference *(applying the brand)* happens "online" in real time with little compute, one image at a time, and has to be interactive

➔ NVIDIA Developer Blog, "Training vs. Inference"
➔ Google AI Blog, "On-Device Machine Learning"

What Models Can Do

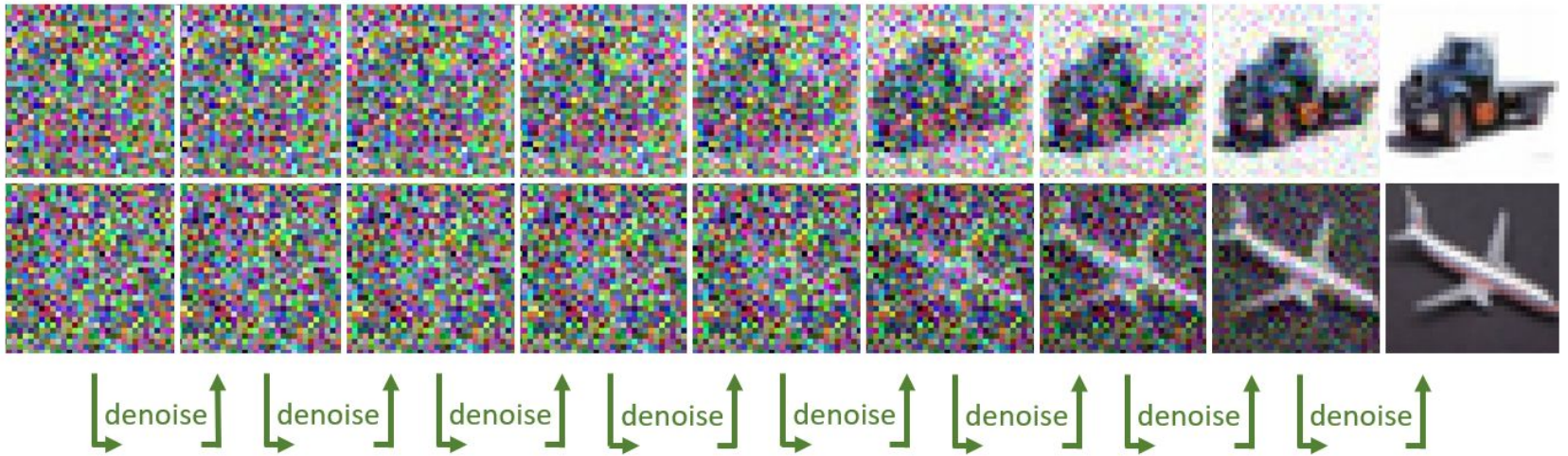
- ➔ YOLO (Redmon et al., 2016)
- ➔ U-Net (Ronneberger, 2015)
- ➔ ResNet (He et al., 2015)
- ➔ ImageNet (Deng et al., 2009)

Generation

Create something new from a description

Examples: DALL·E, Midjourney, Stable Diffusion, generative fill in Photoshop

Produces entirely new images from text or other inputs



Classification

What am I looking at?

Examples: PlantNet identifying a flower, Google Lens identifying a dog breed

With one image as an input, it chooses the best label for the image

Image model

- **26.54% Tibetan Terrier**
- 5.78% Silky Terrier
- 4.58% Dandie Dinmont

➔ https://www.whatbreedismydog.com/

Segmentation

Cut out every object precisely

Example: Remove Background in Figma or Photoshop

Label every pixel in an image to indicate where a subject ends and the background begins

Object Detection

What's in the scene, and where?

Combines **Classification** with **Localization** to identify the location multiple separate objects in the scene, classifying each one

Shipping & The Future

Model Scale & Computational Cost 1/2

Even with a high-end phone like the iPhone 17 Pro Max, you are limited with what models can run on the phone

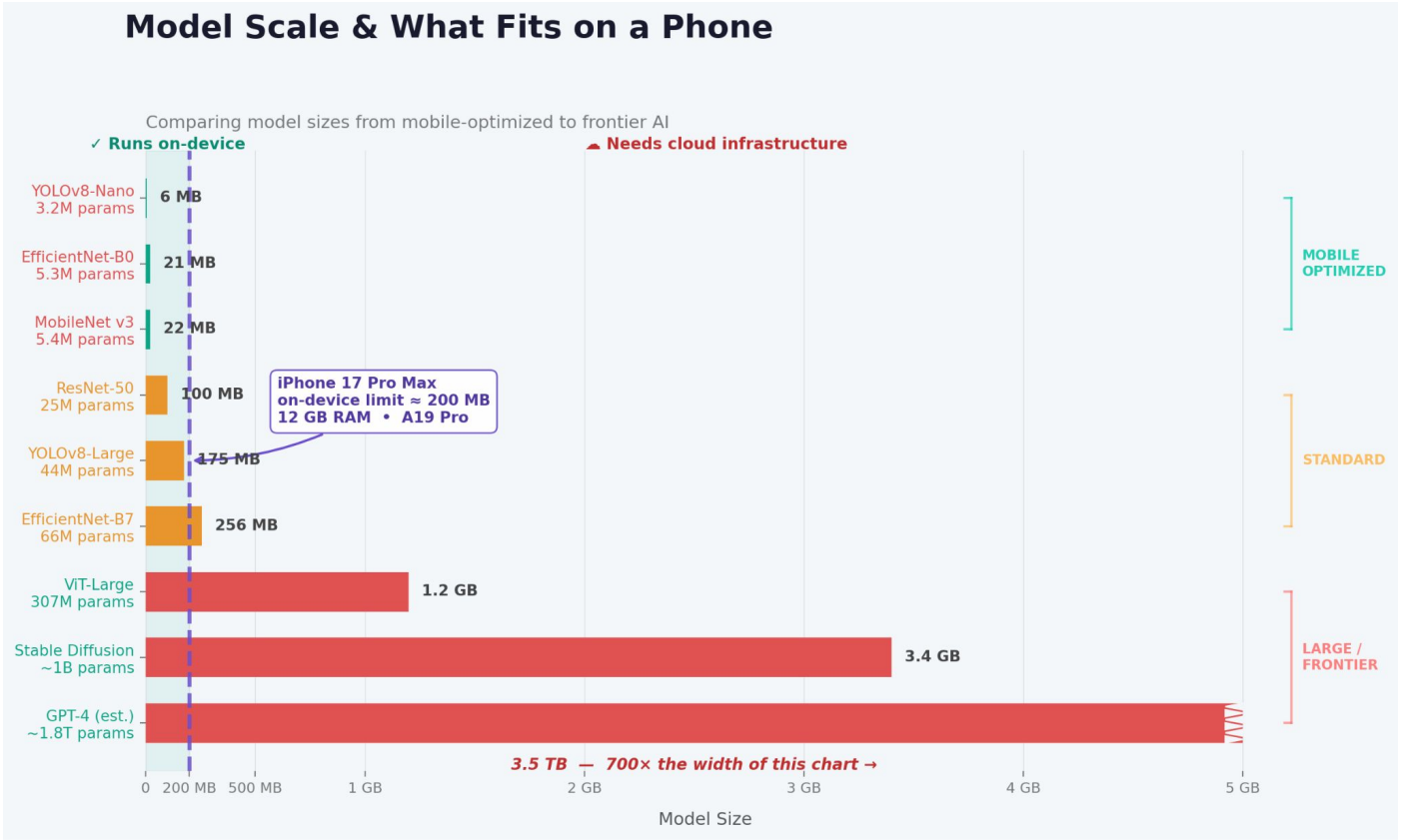
Practical cutoff: 50MB comfortably, 200MB optimized

Optimization techniques:

- **Quantization:** reduce the precision of the model (e.g. only whole numbers)
- **Pruning:** remove unnecessary connections in the model
- **Distillation:** train a small model to act like a big one

➔ Howard et al., "MobileNets" (2017)
➔ Hinton et al., "Distilling the Knowledge" (2015)
➔ Jacob et al., "Quantization and Training of NNs" (2018)
➔ https://www.apple.com/iphone-17/specs/

Model Scale & Computational Cost 2/2



Integrating Image Recognition Into a Mobile App

Architecture: Camera/Gallery → Preprocess → ML Model → Postprocess → Display to user

	On-Device	Cloud
Speed	**Fast (under 50ms)**	Slower (100-500ms)
Privacy	**Images never leave the device**	Images get sent to third party
Needs Signal?	**Works offline**	Yes
Model Size	Very Limited	**Any Size**

- ➔ developer.apple.com/core-ml
- ➔ developers.google.com/ml-kit
- ➔ tensorflow.org/lite
- ➔ mediapipe.dev
- ➔ onnx.ai

Modern AI

Where it's all going

Transformers are changing everything: originally built for language in order to understand all context at once, they are being applied to vision too

Diffusion Models power AI image generation: start with noise and refine it into a real image based on the prompt

Multimodal models combine language and vision models to seamlessly handle text and image input/output

What this means for mobile: these sophisticated models are too large to run on phones, but can be accessed by API. However, mobile devices are being built to run larger models faster

- ➔ Vaswani et al., "Attention Is All You Need" (2017)
- ➔ Dosovitskiy et al., "ViT" (2020)
- ➔ Radford et al., "CLIP" (2021)
- ➔ Rombach et al., "Stable Diffusion" (2022)

Key Takeaways

You don't need to train from scratch. Transfer learning with pre-trained models can get 90%+ accuracy with 100-1,000 images.

On-device ML is becoming more accessible. Phones are being built to run ML models, and you can train models yourself in your browser with no code.

Model size determines deployment strategy. Small models on device, Large on cloud

Small teams can build this. 1-2 people can prototype an image recognition app in weeks with pre-trained models and platform tools

The field moves fast. New types of models are constantly pushing what is possible and many new tools appearing in e.g. Adobe products are built on these advancements

Resources & Further Reading

Learning ML & Computer Vision: cs231n.stanford.edu (Stanford CNN course), fast.ai (practical deep learning, free), 3Blue1Brown (neural network visualizations, YouTube), deeplearning.ai (Andrew Ng's courses), distill.pub (beautiful interactive ML explanations).

Tools & Frameworks: pytorch.org, huggingface.co (model hub), teachablemachine.withgoogle.com (no-code), developer.apple.com/core-ml, developers.google.com/ml-kit.

Key Papers: Goodfellow et al., "Deep Learning" (MIT Press, 2016). He et al., "Deep Residual Learning" (2015). Vaswani et al., "Attention Is All You Need" (2017). Dosovitskiy et al., "An Image is Worth 16x16 Words" (2020). Howard et al., "MobileNets" (2017). Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models" (2022). Radford et al., "Learning Transferable Visual Models" (2021).