Using a CNN model with Flutter requires converting and optimizing your trained model for mobile deployment. Here's what you need to know:

**Model Conversion and Optimization**

**TensorFlow Lite** is the primary solution for Flutter:

* Convert your trained model (PyTorch, TensorFlow, etc.) to TensorFlow Lite (.tflite format)
* Use quantization to reduce model size (float32 to int8 can reduce size by 4x)
* Optimize for mobile inference speed and memory usage

**Model size considerations**:

* Mobile apps should ideally keep models under 10-50 MB
* Larger models increase app download size and device storage usage
* Consider model pruning or knowledge distillation for smaller models

**Flutter Integration**

**TensorFlow Lite Flutter plugin**:

dependencies:

tflite\_flutter: ^0.10.4

tflite\_flutter\_helper: ^0.3.1

**Key implementation steps**:

* Load the .tflite model as an asset in your Flutter app
* **Preprocess images to match training input format (resize, normalize)**
* Run inference using the TensorFlow Lite interpreter
* Post-process outputs to get classification results

**Device Requirements**

**Minimum hardware**:

* RAM: 2-4 GB (depends on model size and complexity)
* Storage: Model size + app overhead
* CPU: Modern ARM processors handle most CNN inference well

**Performance considerations**:

* Inference time: Simple models run in 10-100ms, complex models may take 200-500ms
* Battery usage: Frequent inference can drain battery quickly
* Consider using device GPU acceleration where available

**Development Workflow**

1. Train your CNN model in Python (TensorFlow/PyTorch)
2. Convert to TensorFlow Lite format with quantization
3. Test the converted model's accuracy and performance
4. Integrate into Flutter app using tflite\_flutter plugin
5. Test on actual devices for performance validation

**Alternative Approaches**

**Cloud inference**: Send images to a server API for classification (requires internet) **Edge AI chips**: Some newer phones have dedicated AI processing units **Platform-specific solutions**: Core ML for iOS, ML Kit for cross-platform

The tflite\_flutter plugin is your best bet for offline, cross-platform CNN inference in Flutter apps.