Use Case of TensorFlow Lite: “For This Photo” at VSCO

Mijin An
meeeeejin@gmail.com

VSCO

- VSCO is a photo and video editing app
- It contains a catalog of over 160 presets

Problem Definition

- Users are **overwhelmed by the number of presets** (> 160)
- Users are **stuck to using the few familiar favorites** instead of trying new presets

- VSCO's challenge was to overcome **decision fatigue** by providing trusted guidance and encouraging discovery

The Solution

- VSCO decided to suggest presets for images with on-device ML using **deep CNN models**
  - To make categorization easier and faster than traditional computer vision algorithms

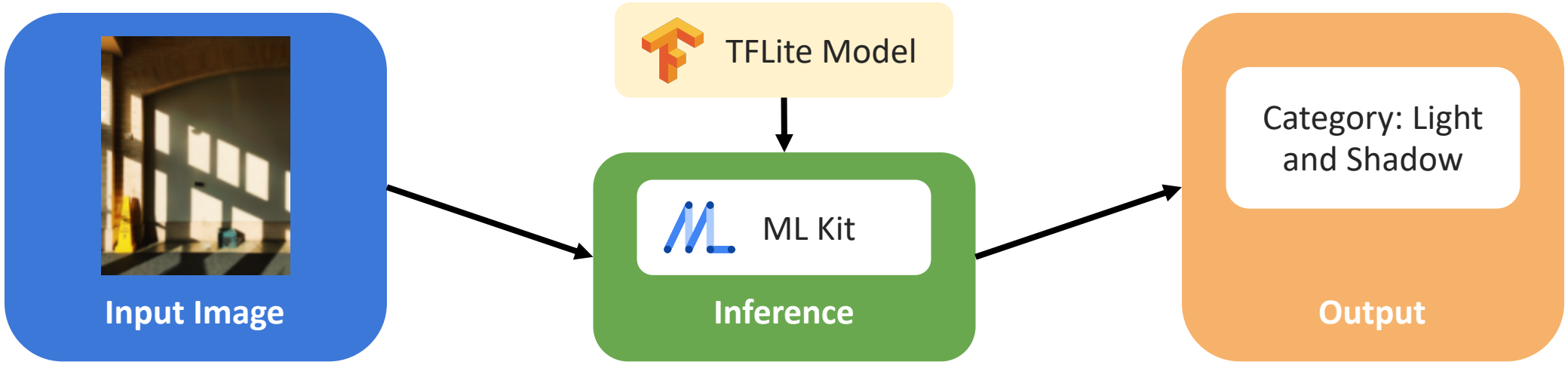
“For This Photo”: Introduction

- This feature uses on-device ML
  - To identify what kind of photo someone is editing
  - Then, to suggest relevant presets from a curated list

1. When a user loads the image in Edit View, inference via the model is instantly kicked off
2. The model returns a category for the image
3. The category ID is then matched with the ID in the cached catalog
4. A list of presets for that category is retrieved
5. Six presets are picked (3 for free+3 for membership)

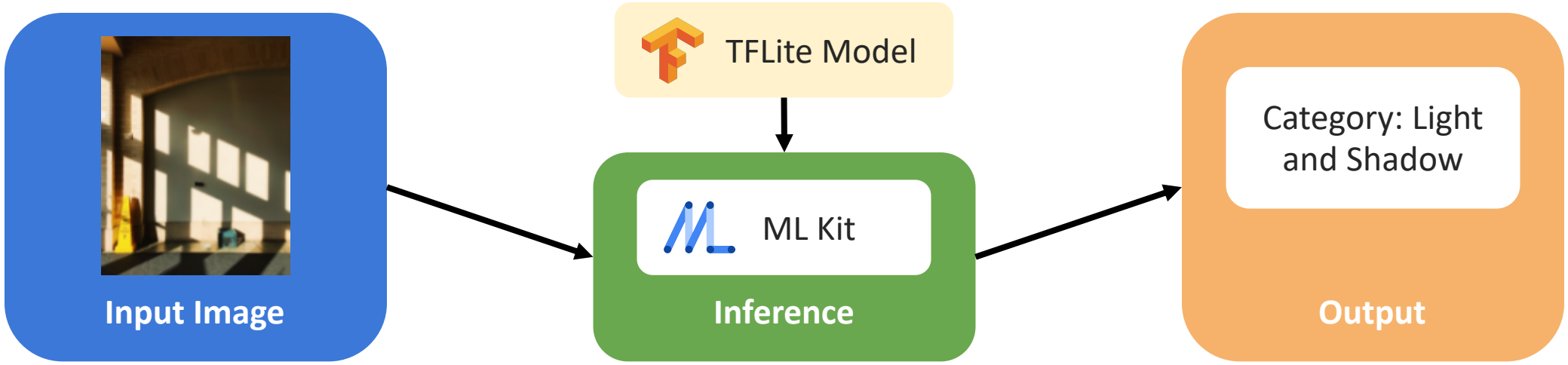
“For This Photo”: How It Works (1)

- Categorizing images
  1. Tag the image data (e.g., portrait, nature, light, shadow, …)
  2. Using the categorized dataset, train a CNN model in TensorFlow based on *SqueezeNet* architecture
     - Because of its smaller size without much loss in accuracy



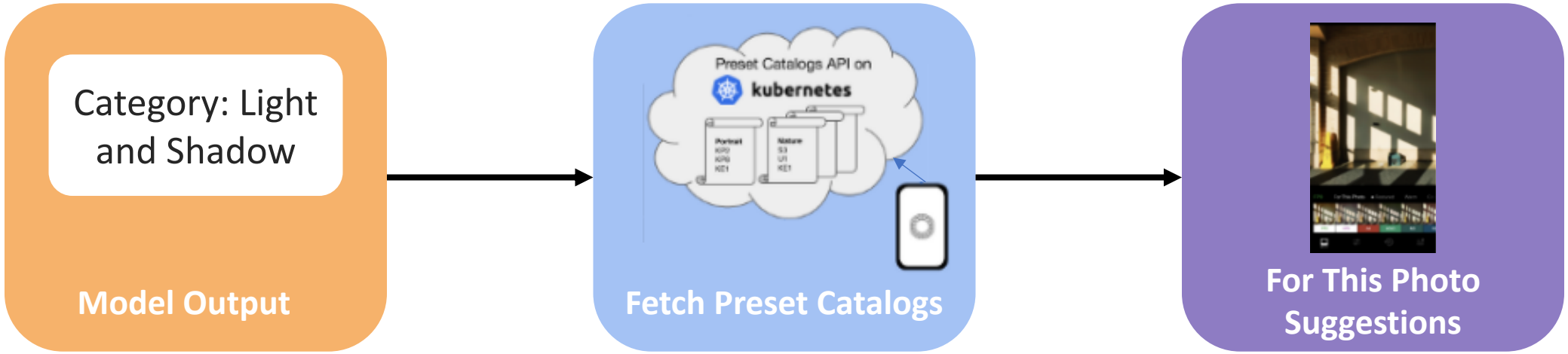
“For This Photo”: How It Works (2)

- Categorizing images
  3. Convert this trained model from TensorFlow’s Saved Model format to TensorFlow Lite (.tflite) format using the *TFLiteConverter*
  4. Bundle it into VSCO app and run inference on images using ML Kit
     - For better accuracy, forgo the quantization step in model conversion and use floating point model



“For This Photo”: How It Works (3)

- Suggesting presets
  1. Collaborate with in-house Imaging team who had created the presets to derive a list of presets that fit the images in each of these categories
  2. Complete the curated catalog with presets mapped to each category



Why On-Device ML?

- **Offline editing**
  - Not to limit users' creativity regardless of the network connectivity
  - Not everyone always has access to high-speed internet
- **Speed**
  - To offer the feature locally, quickly, with no connection required
  - In the case of cloud, it needs time, bandwidth, and data transfer
- **Privacy**
  - A server-side solution cannot access the users' photo before they publish them

Why TensorFlow Lite?

- The **ease** of taking a model trained on the server and converting it to a model compatible for the mobile

- **ML Kit** providing higher level APIs which make the development process much faster
  - Instead, developers can use more time to hone the model

Conclusion

- With on-device ML, VSCO accomplished its goal to make editing with presets easier to navigate by “For This Photo”
- VSCO can help more users not just discover new presets, but zero in on those presets that best matched what they were working on
- VSCO are continuing to invest in the on-device ML area and build more features leveraging this technology in the future

Reference

[1] “Suggesting Presets for Images: Building: “For This Photo” at VSCO.” Medium/TensorFlow. last modified Jun 25, 2019, accessed Sep 3, 2020, https://medium.com/tensorflow/suggesting-presets-for-images-building-for-this-photo-at-vsco-9b94041c4ba4.

[2] “Case Studies and Mentions | TensorFlow.” TensorFlow. accessed Sep 3, 2020, https://www.tensorflow.org/about/case-studies