IESTI05 – Edge AI

Machine Learning System Engineering

3. Image Classification: Introduction

UNIFEI

Marcelo Rovai
Professor Honoris Causa

What is Image Classification?

Definition

Image classification is a fundamental task in computer vision that involves **categorizing an image into one of several predefined classes.**

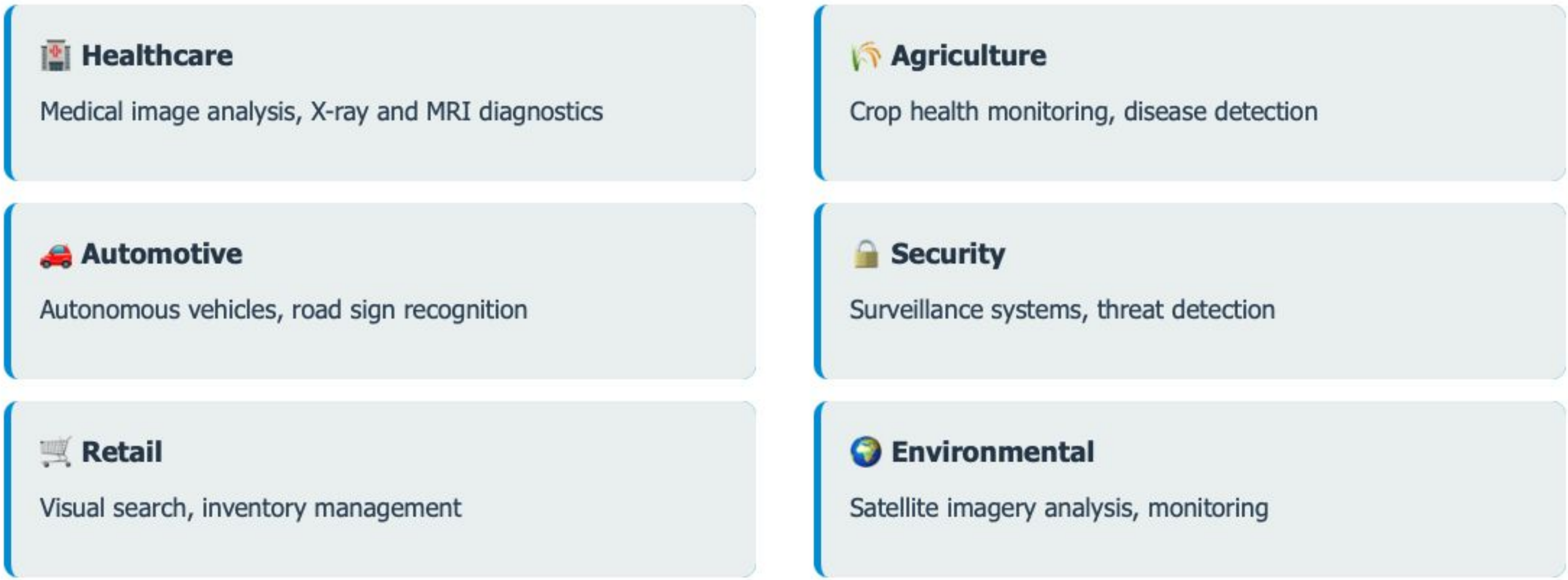
Key Characteristics

- Assigns a single label to the entire image
- Based on visual content analysis
- Uses machine learning algorithms
- Mimics human visual perception

Process Overview

Input Image → Feature Extraction → Classification Model → Predicted Label

Real-World Applications



🏥 Healthcare

Medical image analysis, X-ray and MRI diagnostics

🌾 Agriculture

Crop health monitoring, disease detection

🚗 Automotive

Autonomous vehicles, road sign recognition

🔒 Security

Surveillance systems, threat detection

🛒 Retail

Visual search, inventory management

🌍 Environmental

Satellite imagery analysis, monitoring

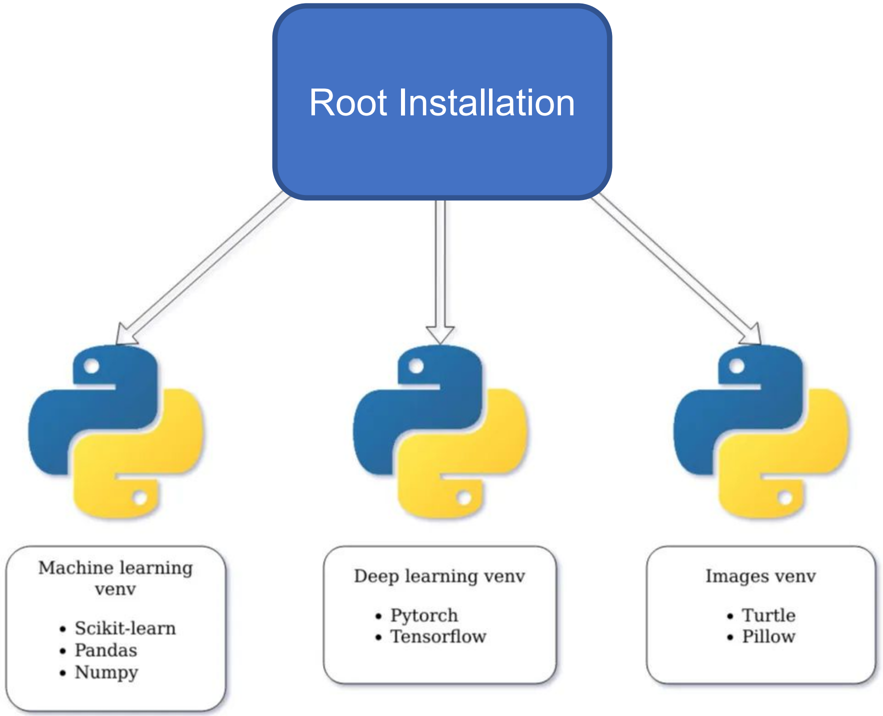
Setting up a Virtual Environment

Activate the environment:

```
source ~/tflite_env/bin/activate
```

To exit the virtual environment, use:

```
deactivate
```



TensorFlow Lite Setup

What is TensorFlow Lite?
A lightweight runtime for running machine learning models on mobile and embedded devices, optimized for inference with minimal latency.

We'll use the TensorFlow Lite runtime for Raspberry Pi, a simplified library for running machine learning models on mobile and embedded devices, without including all TensorFlow packages.

```
pip install tflite-runtime
```

tflite_runtime-2.14.0-cp311-cp311-manylinux_2_34_aarch64.whl

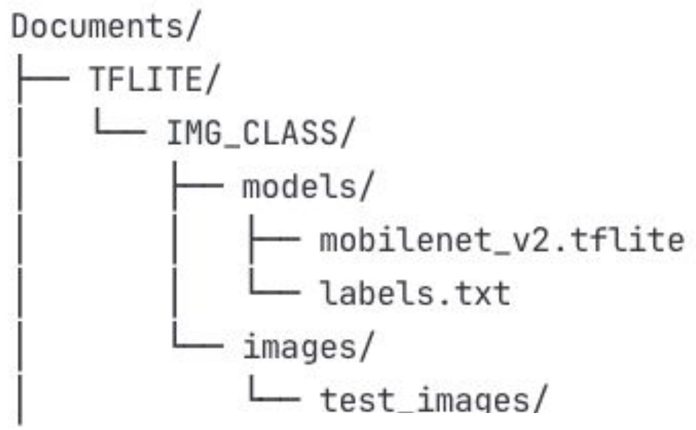
```
sudo reboot
```

Verify installation:

```
pip list | grep -E "(tflite-runtime)"
```

tflite-runtime 2.14.0.

Creating a working directory & get the model



```
Documents/
├── TFLITE/
│   └── IMG_CLASS/
│           ├── models/
│           │   ├── mobilenet_v2.tflite
│           │   └── labels.txt
│           └── images/
│               └── test_images/
```

```
                 test_image.jpeg
```

Go to the models/ folder and get the Model and labels

```
wget
https://storage.googleapis.com/download.tensorflow.org/models/tflite_11_05_08
/mobilenet_v2_1.0_224_quant.tgz
```

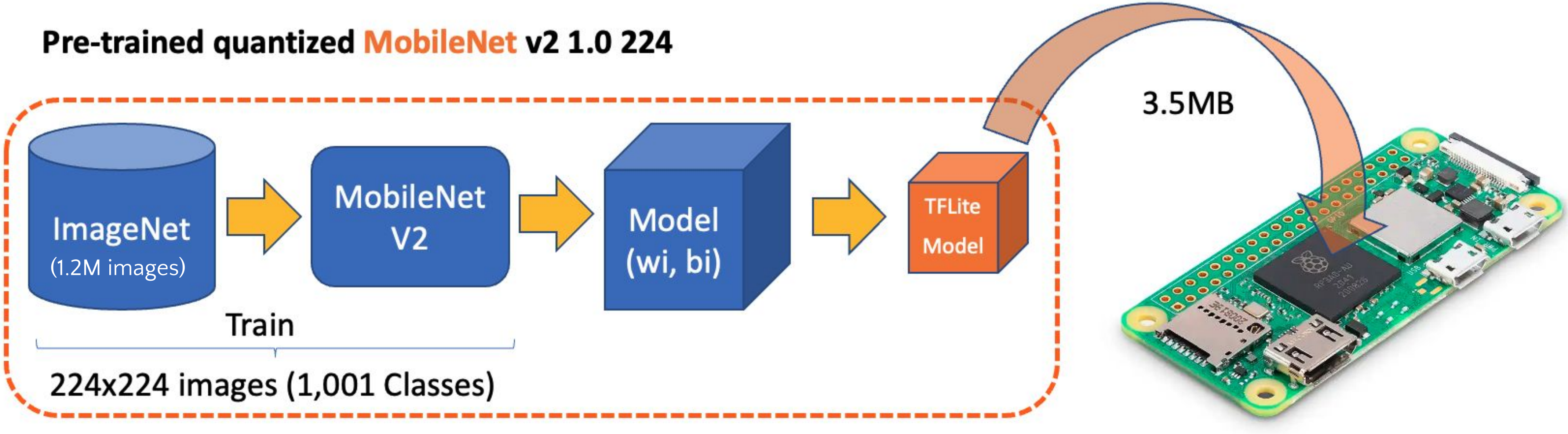
```
tar xzf mobilenet_v2_1.0_224_quant.tgz
```

```
wget
https://raw.githubusercontent.com/Mjrovai/EdgeML-with-Raspberry-Pi/refs/heads
/main/IMG_CLASS/models/labels.txt
```

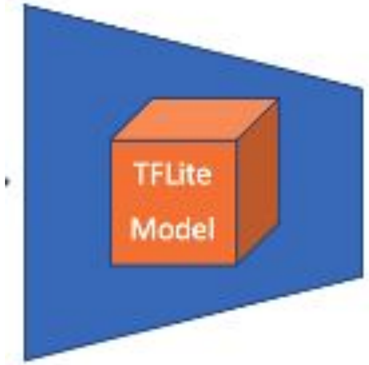
Verifying the Setup

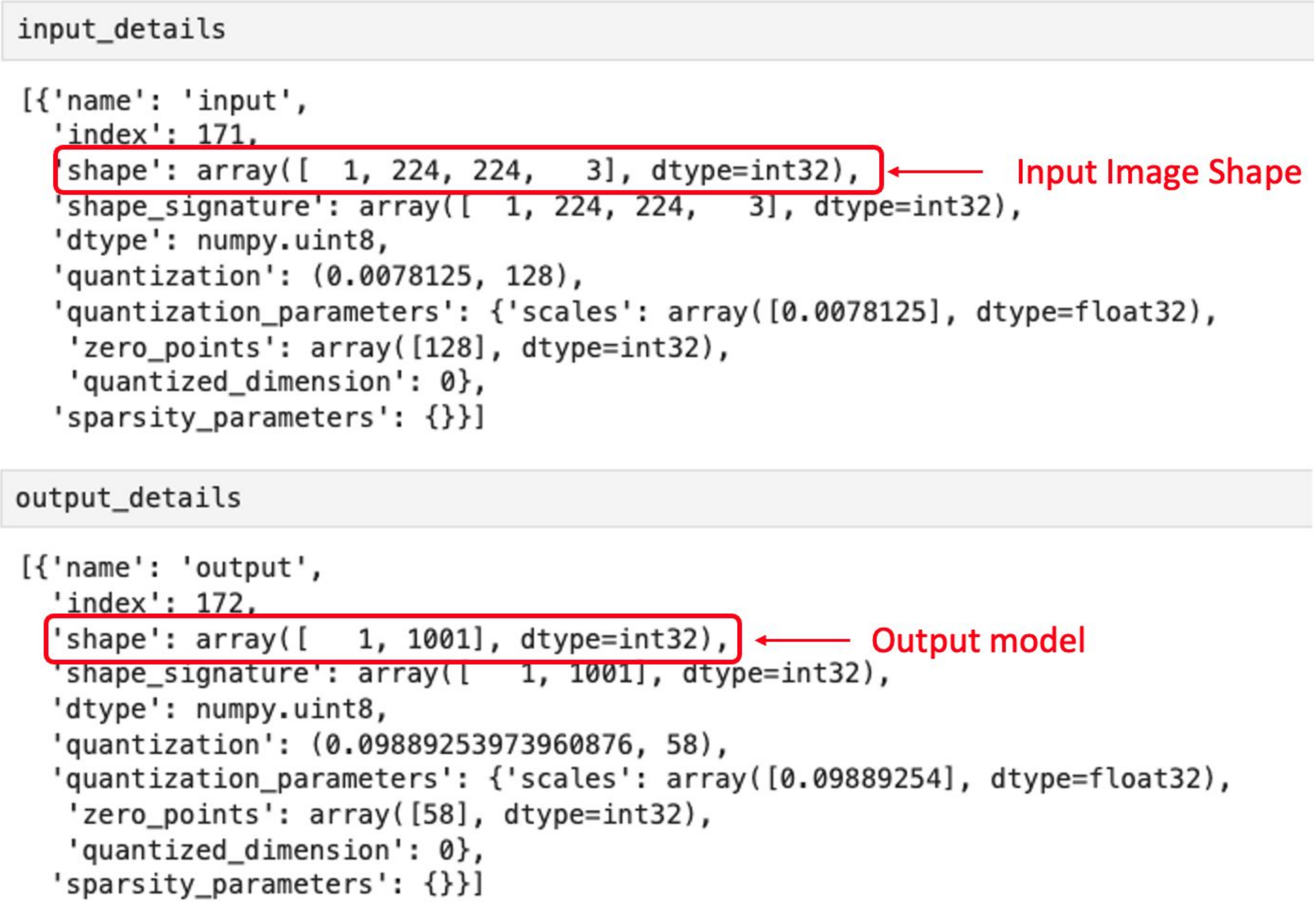
```
import tflite_runtime.interpreter as tflite
import numpy as np
from PIL import Image

print("NumPy:", np.__version__)
print("Pillow:", Image.__version__)

# Try to create a TFLite Interpreter
model_path = "./models/mobilenet_v2_1.0_224_quant.tflite"
interpreter = tflite.Interpreter(model_path=model_path)
interpreter.allocate_tensors()
print("TFLite Interpreter created successfully!")
```

The MobileNet V2 model


Pre-trained quantized MobileNet v2 1.0 224
ImageNet
(1.2M images)
MobileNet
V2
Model
(wi, bi)
TFLite
Model
Train
224x224 images (1,001 Classes)
3.5MB

TFLite Interpreter





```
input_details
```

```
[{'name': 'input',
  'index': 171,
  'shape': array([  1, 224, 224,   3], dtype=int32),
  'shape_signature': array([  1, 224, 224,   3], dtype=int32),
  'dtype': numpy.uint8,
  'quantization': (0.0078125, 128),
  'quantization_parameters': {'scales': array([0.0078125], dtype=float32),
   'zero_points': array([128], dtype=int32),
   'quantized_dimension': 0},
  'sparsity_parameters': {}}]
```

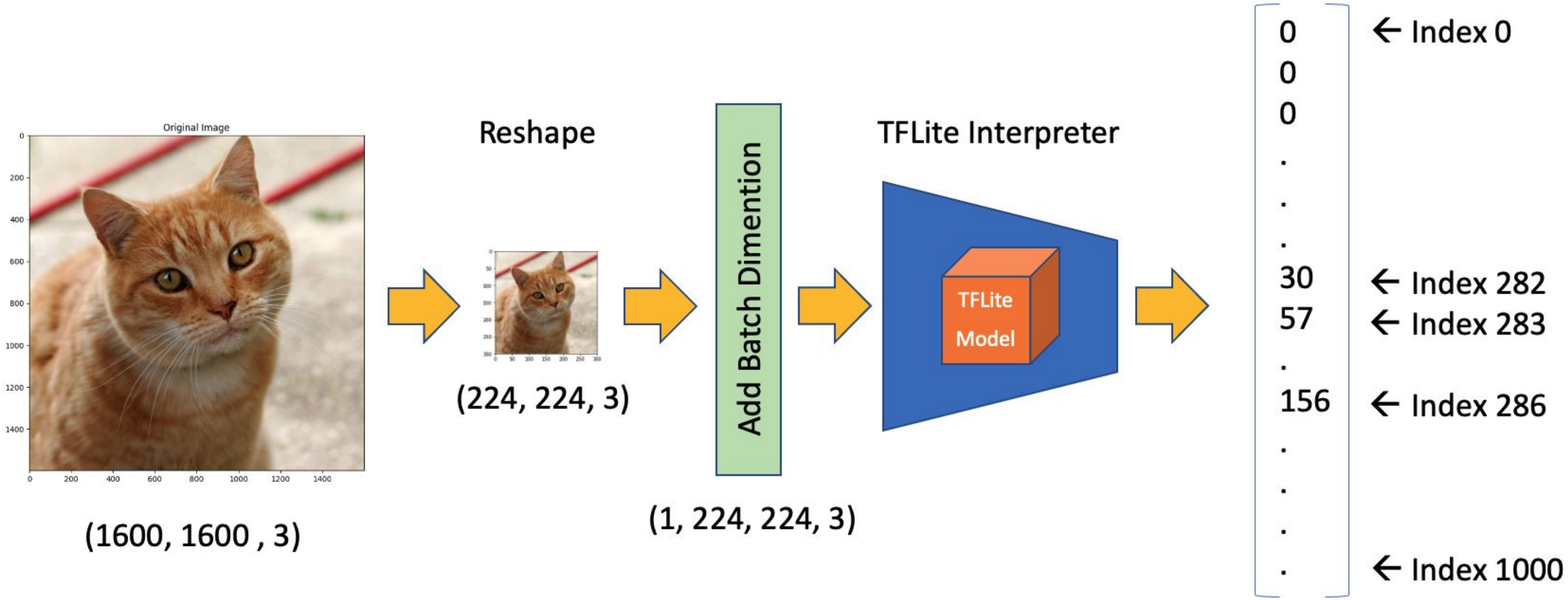
Input Image Shape

```
output_details
```

```
[{'name': 'output',
  'index': 172,
  'shape': array([   1, 1001], dtype=int32),
  'shape_signature': array([   1, 1001], dtype=int32),
  'dtype': numpy.uint8,
  'quantization': (0.09889253973960876, 58),
  'quantization_parameters': {'scales': array([0.09889254], dtype=float32),
   'zero_points': array([58], dtype=int32),
   'quantized_dimension': 0},
  'sparsity_parameters': {}}]
```

Output model

Making inferences with MobileNet V2



10_Image_Classification.ipynb

Making inferences: Static Images and Camera

```
[PREDICTION]           [Prob]

tiger cat              : 37%
Egyptian cat           : 27%
tabby                  : 16%
lynx                   : 10%
carton                 : 2%
```

```
[PREDICTION]           [Prob]

coffee mug             : 99%
cup                    : 0%
whiskey jug            : 0%
teapot                 : 0%
water jug              : 0%
```

Questions?

Prof. Marcelo J. Rovai

rovai@unifei.edu.br

UNIFEI