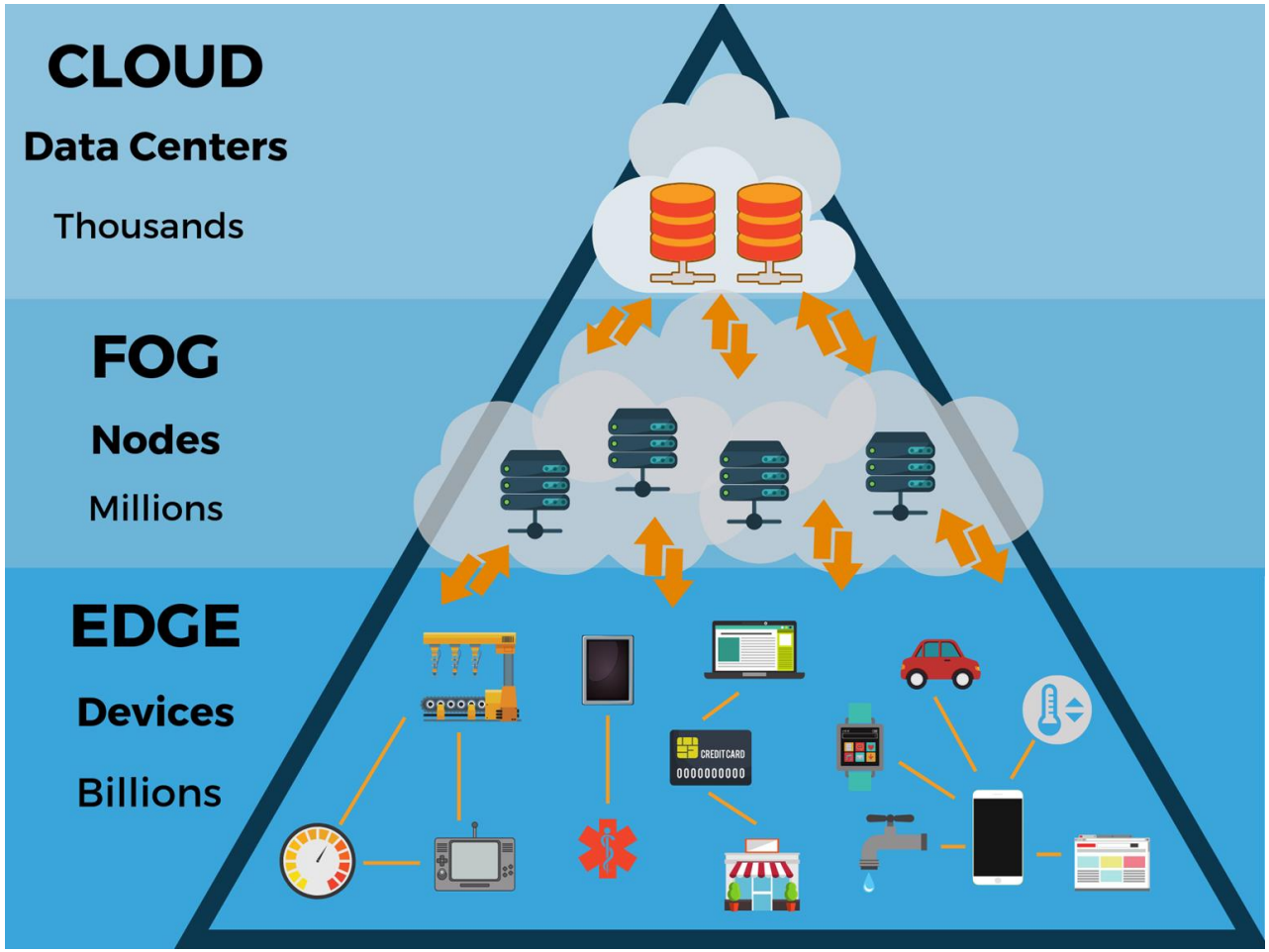
Ubiquitous and Mobile Computing
CS 528: *Tech Talk*

Lokesh Gangaramaney, Kevin Fortier,
Christopher Micek, Andrew Nolan,
Joe Petiti

Computer Science Dept.
Worcester Polytechnic Institute (WPI)

Edge Computing For Machine Learning

What is Edge Computing ?



Historical Context

- Content Delivery Networks were first introduced in the early 1990's
- In 2012, the term fog computing was introduced as a term for dispersed cloud resources.
- Modern use of edge devices- producers of data and not just consumers.

Problem

Proliferation of data and data intensive applications, as well as the increase in computational power on edge devices has put the bottleneck in the network traffic.

ML Applications Require Low Latency

Use Cases

Mobile:

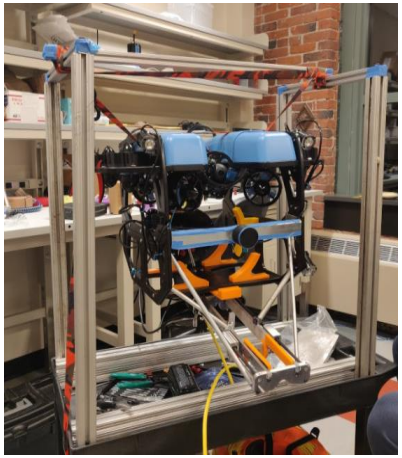
- Facial recognition/detection
- Voice Recognition
- Activity Recognition
- OCR

Robotics and IOT

- Autonomous Vehicles

Applications

- Lionfish MQP
- Pytorch and Tensorflow Mobile Support
  - Image Classification Example



TensorFlow

How It Works

Compute occurs on device

Cloud

Data/Model

Data

Device

Pytorch Image Classification 1

Load Model + Select Image

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    try {
        bitmap = BitmapFactory.decodeStream(getAssets().open("kitten.jpg"));
        // loading serialized torchscript module from packaged into app android asset model.pt,
        // app/src/model/assets/model.pt
        module = Module.load(assetFilePath(this, "model.pt"));
    } catch (IOException e) {
        Log.e("PTMobileWalkthru", "Error reading assets", e);
        finish();
    }

    // showing image on UI
    ImageView imageView = findViewById(R.id.imageView);
    imageView.setImageBitmap(bitmap);
```

Pytorch Image Classification 2

Perform Inference

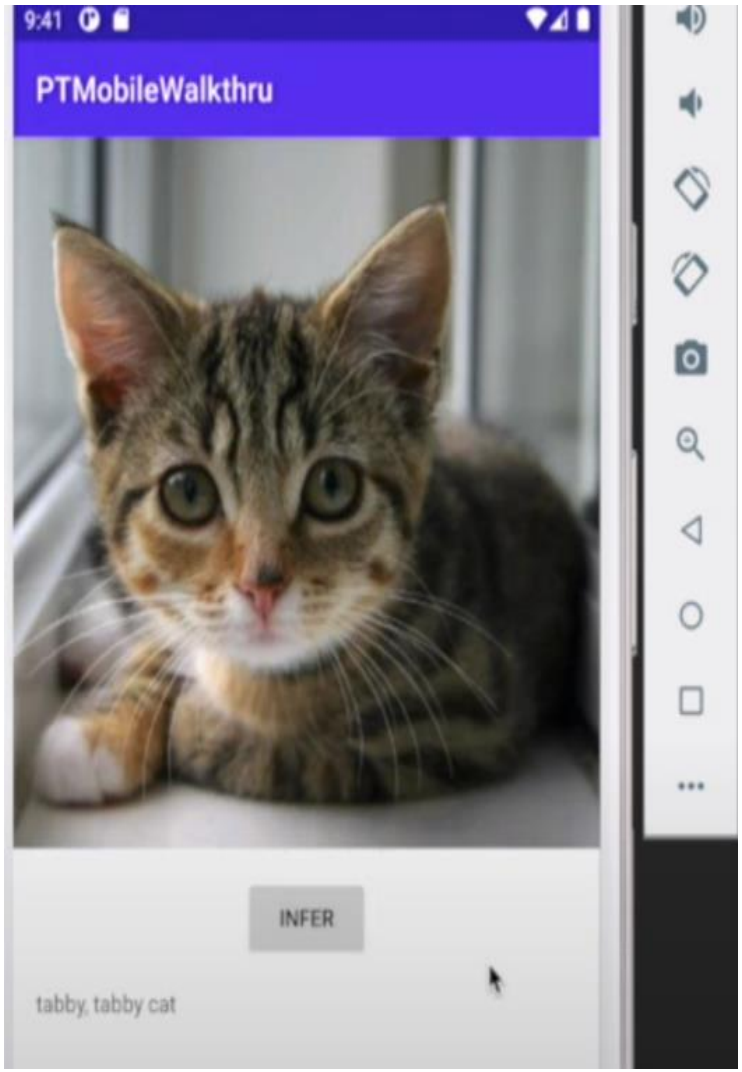
```
final Button button = findViewById(R.id.inferButton);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        final Tensor inputTensor = TensorImageUtils.bitmapToFloat32Tensor(bitmap,
                TensorImageUtils.TORCHVISION_NORM_MEAN_RGB, TensorImageUtils.TORCHVISION_NORM_STD_RGB);

        // running the model
        final Tensor outputTensor = module.forward(IValue.from(inputTensor)).toTensor();

        // getting tensor content as java array of floats
        final float[] scores = outputTensor.getDataAsFloatArray();

        // searching for the index with maximum score
        float maxScore = -Float.MAX_VALUE;
        int maxScoreIdx = -1;
        for (int i = 0; i < scores.length; i++) {
            if (scores[i] > maxScore) {
                maxScore = scores[i];
                maxScoreIdx = i;
            }
        }
```

Pytorch Image Classification 3

Select Class Name

```
public class ImageNetClasses {
  public static String[] IMAGENET_CLASSES = new String[]{
      "tench, Tinca tinca",
      "goldfish, Carassius auratus",
      "great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias",
      "tiger shark, Galeocerdo cuvieri",
      "hammerhead, hammerhead shark",
      "electric ray, crampfish, numbfish, torpedo",
      "stingray",
      "cock",
      "hen",
      "ostrich, Struthio camelus",
      "brambling, Fringilla montifringilla",
      "goldfinch, Carduelis carduelis",
      "house finch, linnet, Carpodacus mexicanus",
      "junco, snowbird",
      "indigo bunting, indigo finch, indigo bird, Passerina cyanea",
      "robin, American robin, Turdus migratorius",
      "bulbul",
      "jay",
      "magpie",
```

Pytorch Image Classification 4



References

J. Chen and X. Ran, “Deep Learning With Edge Computing: A Review,” Proc. IEEE, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

M. Satyanarayanan. The emergence of edge computing. Computer (Long. Beach. Calif), 50 (2017), pp. 30-39.

Pytorch code tutorial:
https://github.com/pytorch/workshops/tree/master/PTMobileWalkthruAndroid

W. Shi, J. Cao, Q. Zhang, Y. Li. Edge computing: vision and challenges. IEEE Internet Things J., 3 (5) (2016), pp. 637-646.