# Distributed deep neural networks over the cloud, the edge, and end devices

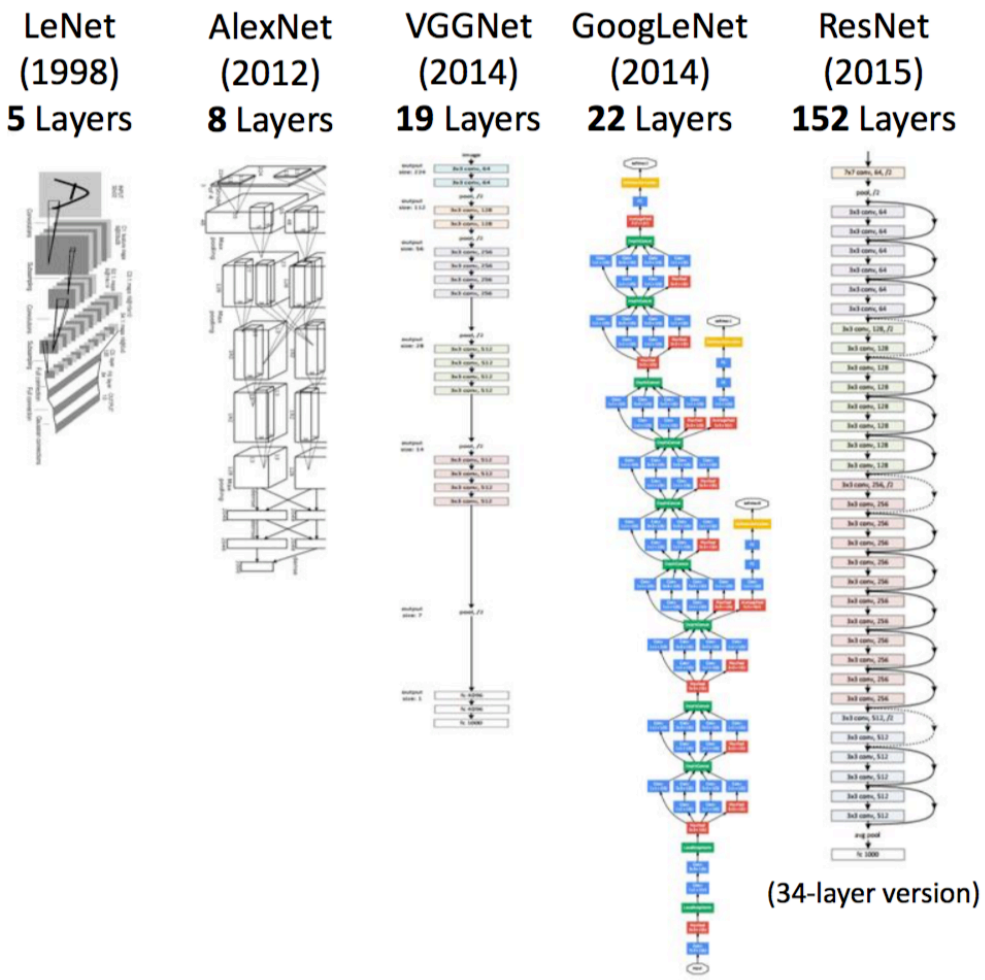
Harvard University

ICDCS 2017

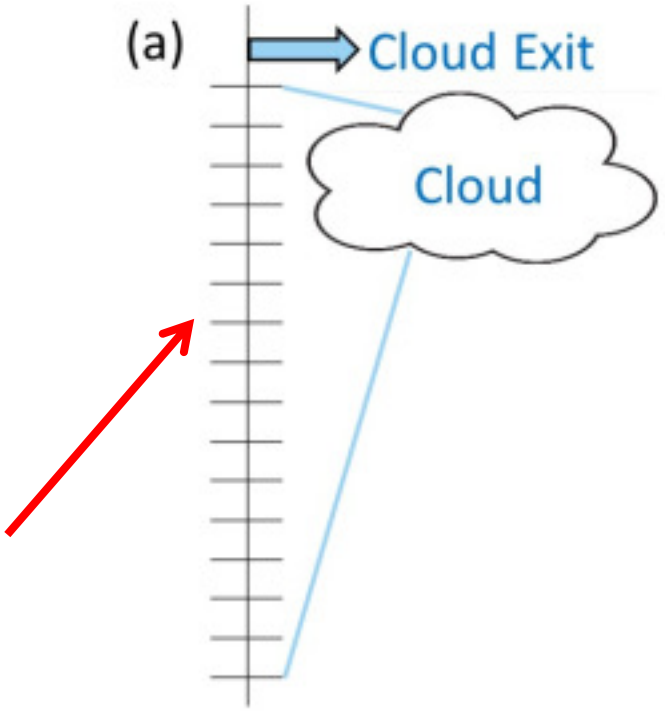
# Goals of DDNN (Distributed Deep Neural Network)

- Achieving a deep neural network over a distributed system
- Scaling up in neural network size and scaling out in geographical span
- Considering sensor fusion, system fault tolerance and data privacy for DNN applications

# Progression towards deeper neural network structures

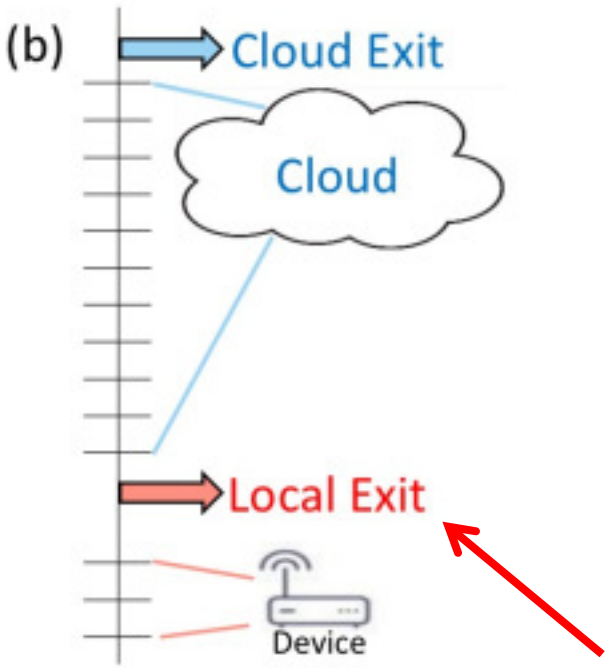


# The Classic typology




- Upload everything to the cloud and process it there.

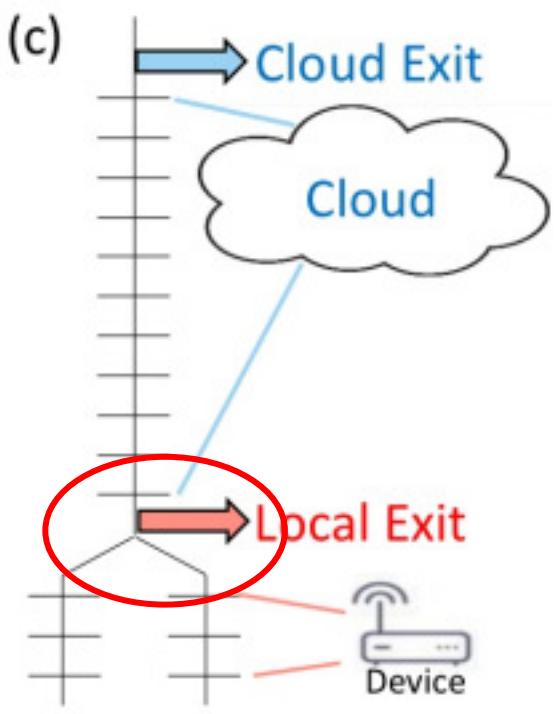
# Considering the local processing




- Do some local processing on device.
- Then transfer a compressed representation to the cloud for further processing

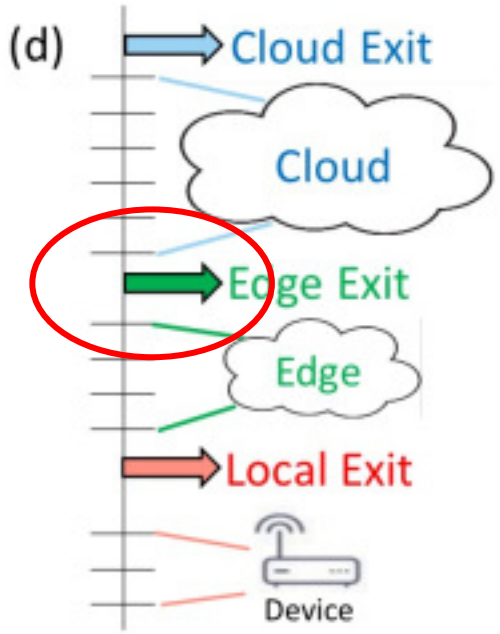
- Local exit: can stop here if we're confident enough.

# Considering horizontal scaling from multiple distributed devices



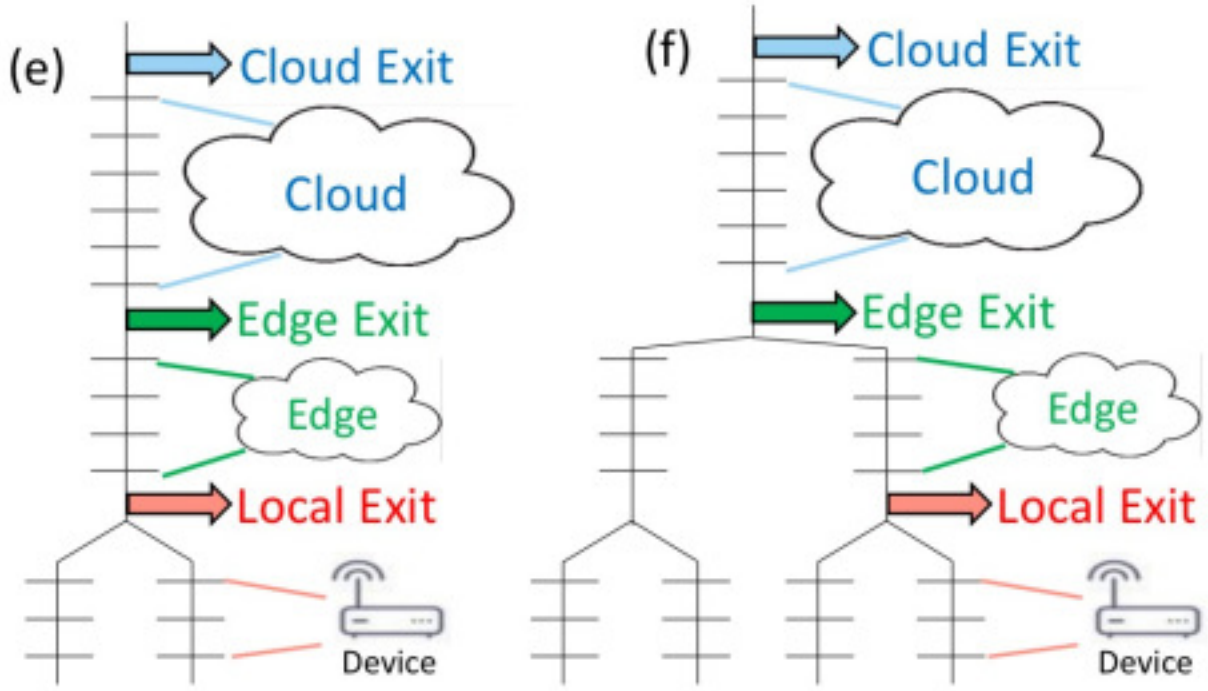

- The device inputs are aggregated in the lower layers of the network.
- Different aggregation algorithms, such as max-pooling (MP), average-pooling (AP), and concatenation (CC)

# Considering the vertical scaling by introducing an additional layer




- With another early exit opportunity at the edge

# Combined horizontal and vertical scaling



# Local Exits

- *Binarized neural networks**
  - weights in linear and convolutional layers are constrained to {-1, 1}
- *Embedded binarized neural networks*
  - fit on embedded devices by reducing floating point temporaries through re-ordering the operations in inference (---minimizing the required memory)

*for less memory and reduced computation

# Training

- Over the distributed computing hierarchy, the DDNN system can be trained on a **single** powerful server or in the cloud
- The loss from each exit **is combined** during back-propagation (described as GoogleNet and BranchyNet)
- The accuracy of each stage is **relative to its depth**.

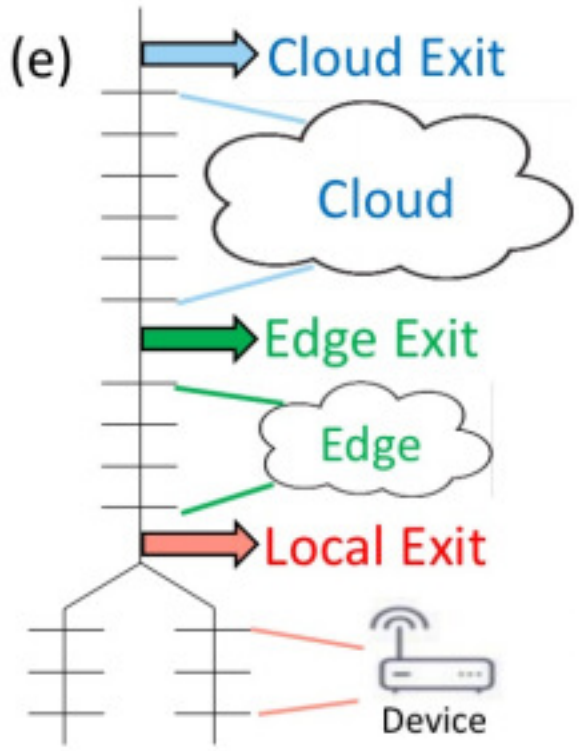
# Inference in stages

- confidence criteria: a normalised entropy threshold (between 0 to 1)

$$\eta(x) = -\sum_{i=1}^{|C|} \frac{x_i \log x_i}{\log |C|}$$

where C is the set of all possible labels, x is a probability vector.

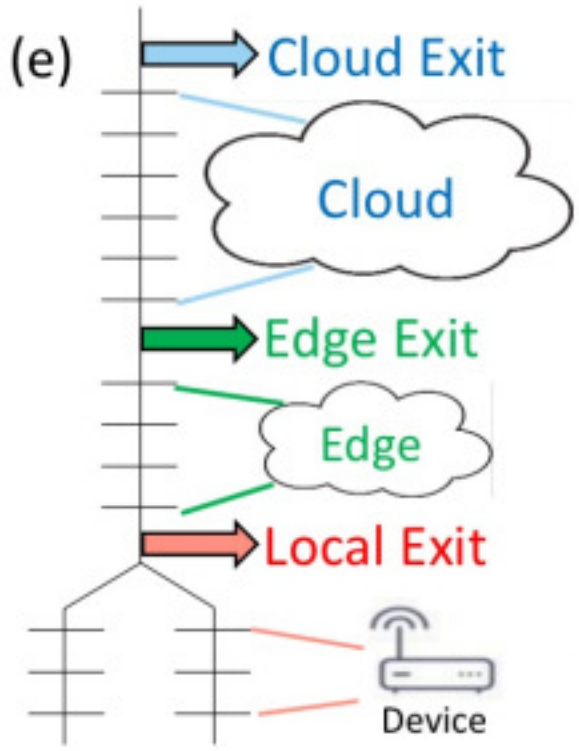
# An illustration




1. Each end device first sends summary information to the local aggregator

2. The local aggregator determines if the combined summary information is sufficient for accurate classification.

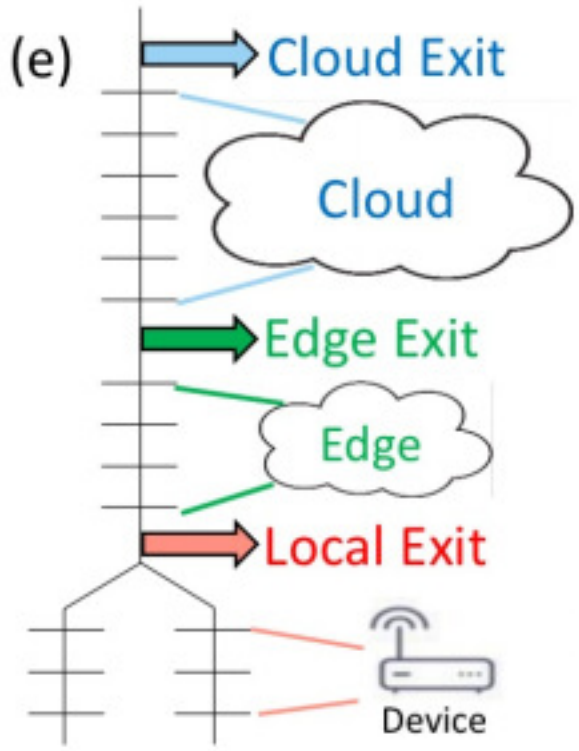
# An illustration




3. If so, the sample is classified (exited).

4. If not, each device sends more detailed information to the edge in order to perform further processing for clarification.

# An illustration



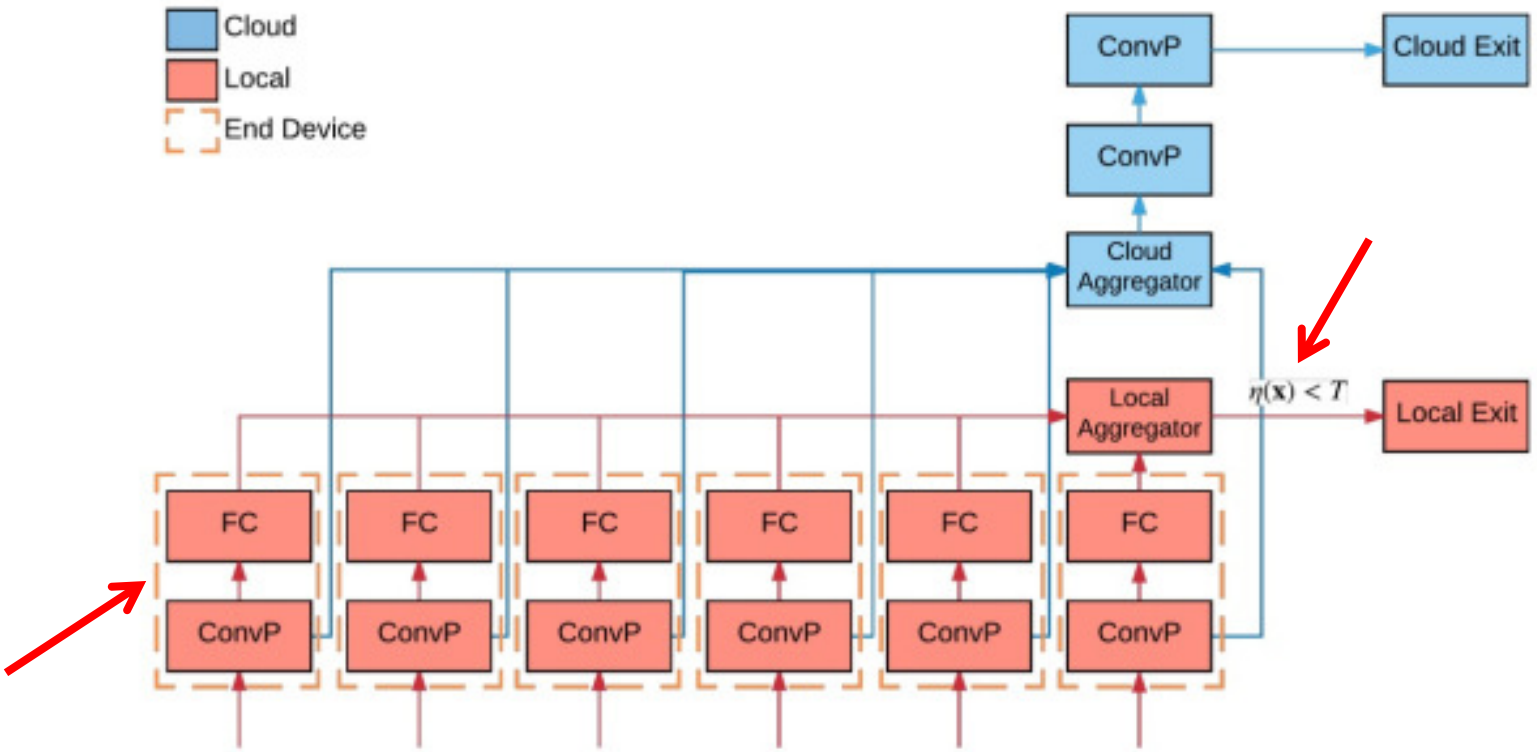

5. If the edge believes it can correctly classify the sample it does so and no information is sent to the cloud.

6. Otherwise, the edge forwards intermediate computation to the cloud which makes the final classification.

# Multi-view, multi-camera scenario


Figure 5. Example images of three objects (person, bus, car) from the multi-view multi-camera dataset. The six devices (each with their own camera) capture the same object from different orientations. An all grey image denotes that the object is not present in the frame.

# The DDNN architecture




ConvP: binary convolution-pool  FC: binary fully connected

There are 680 training samples and 171 testing samples.

Showing the performance according to the number of devices

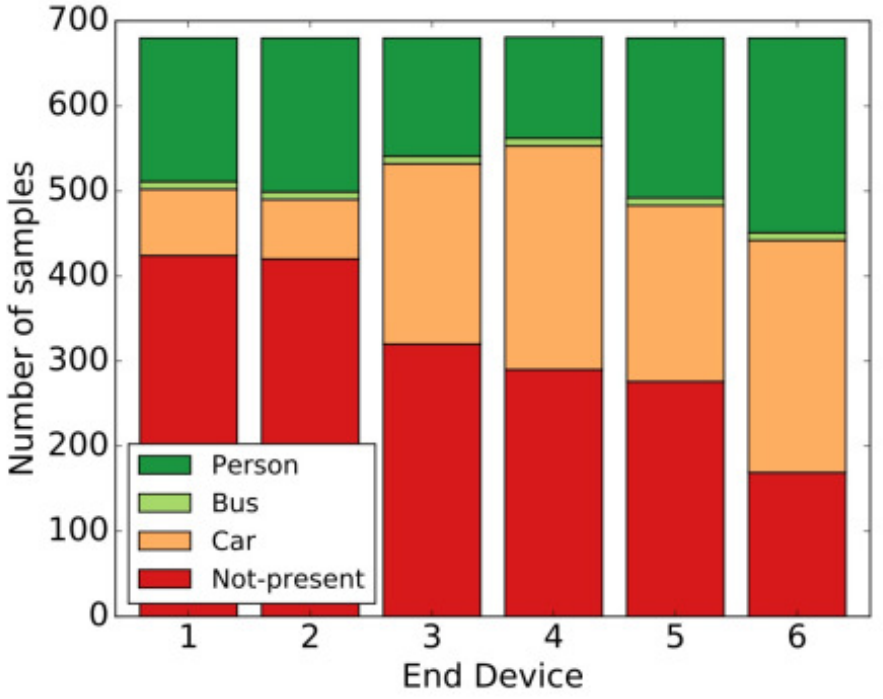



Figure 6. The distribution of class samples for each end device in the multi-view multi-camera dataset.

# Different Schemes

The **best** aggregation combination turns out to be max pooling at the local aggregator, and concatenation at the cloud aggregator

CC: concatenation

at the cloud aggregator, back-propagation gradients are passed through **all devices**

| Schemes | Local Acc. (%) | Cloud Acc. (%) |
|---|---|---|
| MP-MP | 95 | 91 |
| **MP-CC** | **98** | **98** |
| AP-AP | 86 | 98 |
| AP-CC | 75 | 96 |
| CC-CC | 85 | 94 |
| AP-MP | 88 | 93 |
| MP-AP | 89 | 97 |
| CC-MP | 77 | 87 |
| CC-AP | 80 | 94 |

# Different Thresholds

The performance according to different thresholds for local exit

the threshold to 0.8 results in the best overall accuracy with significantly reduced communication

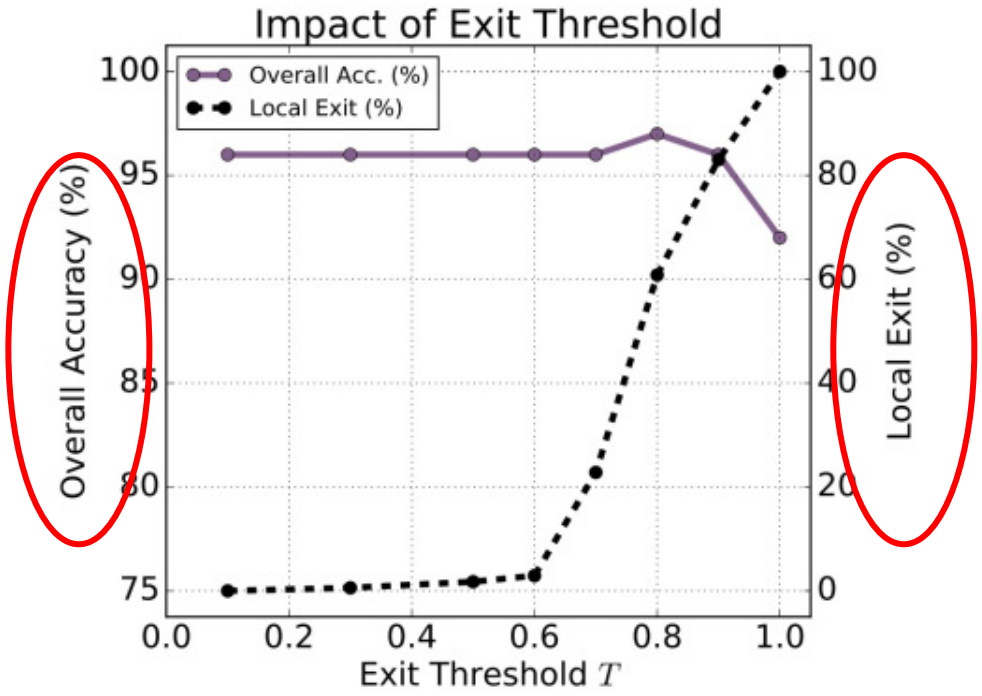



Figure 7. Overall accuracy of the system as the entropy threshold for the local exit is varied from 0 to 1. For this experiment, 4 filters are used in the ConvP blocks on the end devices.

# What we learn.

- For some types of neural networks, it might be possible to divide multiple layers into different pieces, and deploy pieces into different place for achieving kind of distributed hierarchy.

- Local exits can make faster response and reduce the computational overhead in the cloud.