# Design and Implementation of Computational Offloading in Mobile Edge Computing for Augmented Reality Applications

Master thesis

Alex Justesen Karlsen

September 4, 2019

# Preface

> Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.
>
> Alex Justesen Karlsen

# Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

# Acronyms

**AI** Artificial Intelligence. 2

**AR** Augmented Reality. 3

**DDNN** Distributed Deep Neural Network. IV, 3

**EI** Edge Intelligence. 2, 3

**IoT** Internet of Things. 2

**VR** Virtual Reality. 3

**WAN** Wide Area Network. 2, 3

# Contents

# Introduction

Recent years breakthrough within deep learning have led to a dramatic increase in the amount of Artificial Intelligence (AI) applications and services, such as personal assistants, recommendation systems and surveillance systems. Combined with the development of mobile computing and Internet of Things (IoT), where billions of device are getting connected to the internet. Traditionally data for AI applications and services are generated by the devices and transferred to large data centers for computation. To fully unleash the power of AI applications and services Edge Intelligence (EI) or edge AI have become an interesting research area. EI have potential to reduce the

## 0.1 Edge Intelligence

Motivation??

### 0.1.1 Architectures

Inference architectures for edge-centric EI application and services can be categories into four main models [1]:

**Device-based Mode** end device obtain model from edge server. The end-device then acquires input data and performs model inference. Since all computation is done on the end device, performance is solely reliant on the end device's computing resources.

**Edge-based Mode** end device acquires input data. The input data is transferred to an edge server, which performs model inference and send the prediction results to the end device. The performance relies on edge server computing resources and network bandwidth.

**Edge-Device Mode** end device acquires input data and performs partially model inference. The intermediate data is transferred to an edge server which finalizes model inference. The performance relies on end device's and edge server computing resource, network bandwidth and edge server workload.

**Edge-Cloud Mode** resemble edge-device mode, however the model inference task is now partitioned between edge server and cloud data centers. The model is now reliant on edge server and data

center computing resources, but even more reliant on Wide Area Network (WAN) transmission rate between edge and cloud.

### 0.1.2 Performance Metrics

The aim of edge intelligence is to accommodate certain performance metrics:

**Latency** is defined as the overall time of the inference process, including from data is generated at the device, data transmission, preprocessing, model inference and postprocessing. For EI real-time application, such as Augmented Reality (AR)/Virtual Reality (VR), where stringent deadlines requirements must be met e.g. 100ms. Latency is affected by several factors; computing resources, data rate, model architecture and execution.

**Accuracy** is defined as the ratio correctly predicted input samples from the total number of inputs.

$$\alpha = \frac{n_c}{N}$$

Accuracy requirements are dependent on the EI application, for instance autonomous vehicles require extreme accuracy with extremely low latency. The essential trade-off is how accurate a model can we use while still satisfying latency demands.

**Energy efficiency** is important as end devices are typically battery powered. Offloading model inference to edge servers introduces communication overhead for the EI service.

**Communication overhead** is introduced for all modes, except device-based mode, whenever inference is offloaded for remote computation. The cost in terms of latency is dependent network connection to edge servers and even more reliant on unreliable and expensive WAN connection to a cloud data center.

**Privacy** data generated by end devices might be confidential, hence not allowed to be processed by a data center unless confidentiality can be guaranteed. Privacy relies on how data is process within the EI application.

**Memory footprint** description

### 0.1.3 Enabling technologies (related work?)

## 0.2 Distributed Deep Neural Network

Distributed Deep Neural Network (DDNN) is an early exit framework proposed by Teerapittayanon, McDanel, and Kung [2] as their fllow-up on BranchyNet [3] extending the early exit model into a distributed system over cloud, edge and end devices.

This thesis implements the DDNN framework, however extending the device model to implement [4] and edge model to implement ResNet152 [5] and train on a more complex dataset, Pascal VOC [6].

# References


[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing", en, *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019, ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2019.2918951. [Online]. Available: https://ieeexplore.ieee.org/document/8736011/ (visited on 08/22/2019).

[2] S. Teerapittayanon, B. McDanel, and H. Kung, "Distributed Deep Neural Networks Over the Cloud, the Edge and End Devices", en, in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Atlanta, GA, USA: IEEE, Jun. 2017, pp. 328–339, ISBN: 978-1-5386-1792-2. DOI: 10.1109/ICDCS.2017.226. [Online]. Available: http://ieeexplore.ieee.org/document/7979979/ (visited on 08/22/2019).

[3] ——, "BranchyNet: Fast inference via early exiting from deep neural networks", en, in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Cancun: IEEE, Dec. 2016, pp. 2464–2469, ISBN: 978-1-5090-4847-2. DOI: 10.1109/ICPR.2016.7900006. [Online]. Available: http://ieeexplore.ieee.org/document/7900006/ (visited on 08/26/2019).

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", en, *arXiv:1801.04381 [cs]*, Jan. 2018, arXiv: 1801.04381. [Online]. Available: http://arxiv.org/abs/1801.04381 (visited on 08/22/2019).

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", en, *arXiv:1512.03385 [cs]*, Dec. 2015, arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385 (visited on 09/04/2019).

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge", en, *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010, ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-009-0275-4. [Online]. Available: http://link.springer.com/10.1007/s11263-009-0275-4 (visited on 08/22/2019).

# Appendix

# List of Figures

# List of Tables

# Glossary

**deep learning** A discipline within machine learning using neural networks with a large number of hidden layers. 2