SCHOOL OF ENGINEERING SCIENCE

MULTIMEDIA LABORATORY

# DFTS Version 2: User Documentation

TM-ML-2021-001

---

Hans Dhondea
ashivdhondea5@gmail.com
hdhondea@sfu.ca

September 14, 2021

# Contents

# List of Figures

# List of Abbreviations

**ALTeC** Adaptive Linear Tensor Completion. 4

**CALTeC** Content Adaptive Linear Tensor Completion. 4

**CI** Collaborative Intelligence. 4

**DFTS** Deep Feature Transmission Simulator. 1–4

**DNN** Deep Neural Network. 1

**HaLRTC** High accuracy Low Rank Tensor Completion. 4

**SFU** Simon Fraser University. 1

**SiLRTC** Simple Low Rank Tensor Completion. 4

Chapter 1

# Introduction

This technical report serves as user documentation for DFTS version 2. The original DFTS, developed by Harsha at SFU's Multimedia Lab in 2018, had the objective of serving as a testbed for studies in packet-based transmission of deep features over unreliable communication channels [1].

A more sophisticated simulation framework called DFTS2, presented in [2], is now compatible with TensorFlow 2. It has the following new capabilities: additional channel models and missing feature recovery methods from the recent literature. In addition to these, it can compute additional performance metrics such as Top-5 prediction accuracy for the image classification task.

## 1.1 What can DFTS version 2 do?

We will use "DFTS" to refer to DFTS2 from this point onward. Briefly, DFTS can split a pre-trained deep model (DNN) into a mobile sub-model and a cloud sub-model in a collaborative intelligence system. to continue.

include figures.

running dfts from the terminal. show code snippet and example yaml file.

Broadly, there are two types of experiments possible with DFTS: single-shot or "demo‘ experiments and Monte Carlo experiments. Demonstration experiments are meant to be run on a small test set to generate deep feature tensor packet visualizations whereas Monte Carlo experiments are meant to be "full-scale' experiments run on the entire test set to compute overall statistical results. Single-shot experiments run very quickly and produce qualitative results while full-scale experiments may take hours and produce quantitative results.

## 1.2 Overview of this document

- Chapter 2 describes in detail each component of DFTS.
- Chapter 3 explains how to run demonstration experiments.
- Chapter 4 explains how to run Monte Carlo experiments.
- Chapter 5 concludes this technical report and provides recommendations for future work.

Chapter 2

# Simulator description

## 2.1 Introduction

This chapter discusses in detail the different components of DFTS.

## 2.2 Intermediate feature tensor packetization

## 2.3 Channel models

DFTS offers three models for the communication channel model: random loss, i.e., independent and identically distributed (iid), Gilbert-Elliott and external packet traces. Technically, the perfect channel model case, that is, no packet loss at all, is also a channel model. We will next look at each channel model case.

### 2.3.1 Random loss channels

### 2.3.2 Gilbert-Elliott channels

### 2.3.3 Packet traces

## 2.4 Packet loss concealment

By recovering missing packets in deep feature tensors, we are able to provide the cloud sub-model with more “complete‘’ tensor data to exploit in order to complete the inference task. DFTS allows the user to switch on or off error concealment. The following packet loss concealment methods include

- general tensor completion methods: SiLRTC [3] and HaLRTC [3].
  These methods were developed to recover missing values in visual data. They make no assumption as to the nature of the loss (in other words, they are agnostic to the chosen packetization scheme). SiLRTC and HaLRTC adopt an iterative approach: they have to be run a number of times on the corrupted tensor to recover the missing packets. The number of iterations to be run is a parameter for the engineer to tune.

- methods which are specific to deep feature tensors in CI: ALTeC [4] and CALTeC [5].
  ALTeC was originally developed for a single row of feature data per packet scheme. DFTS provides an ALTeC modified to handle a multiple rows of feature tensor data per packet scheme.

- image inpainting-based methods, such as Navier-Stokes [6] which has been used for error concealment in CI in [7].

## 2.5 Simulation mode

Chapter 3

# Demo experiments

## 3.1 Introduction

## 3.2 Quantization demonstration

## 3.3 Lossy channel demonstration

Chapter 4

# Monte Carlo experiments

## 4.1 Introduction

## 4.2 Quantization experiments

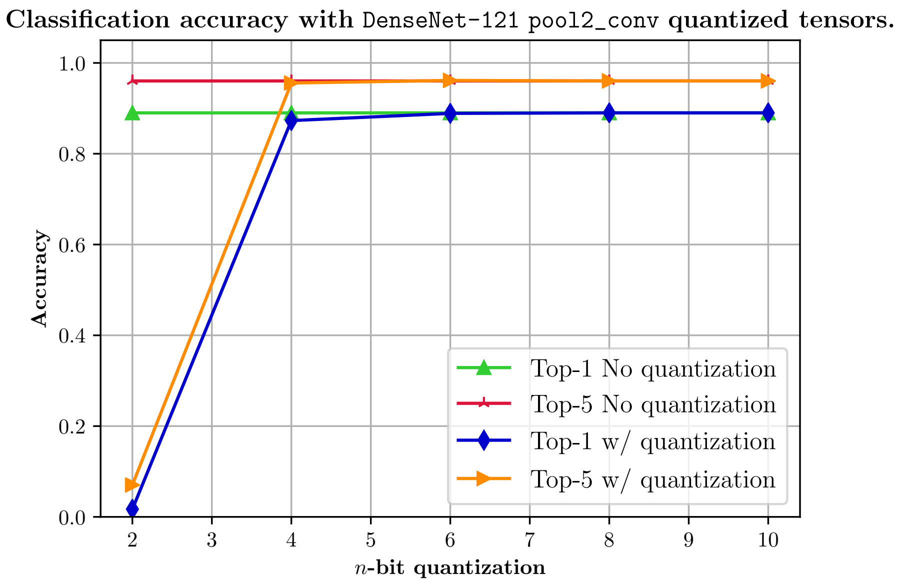


Figure 4-1: Quantization experiments with densenet.

## 4.3 Lossy channel experiments

As discussed in 2.3, there are two options for imperfect channel models: random loss and Gilbert-Elliott channel models. We will focus our discussion on Gilbert-Elliott channel model experiments.

## 4.4 Experiments with external packet traces

## 4.5 Summary

Chapter 5

# Future work

# Bibliography


[1] H. Unnibhavi, H. Choi, S. R. Alvar, and I. V. Bajić, “DFTS: Deep feature transmission simulator.” *IEEE MMSP’18* demo, 2018. https://github.com/SFU-Multimedia-Lab/DFTS. [1]

[2] A. Dhondea, R. A. Cohen, and I. V. Bajić, “DFTS2: Deep feature transmission simulator for collaborative intelligence,” in *Proc. IEEE VCIP*, (Munich, Germany), 2021. to appear. [1]

[3] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, pp. 208–220, Jan. 2013. [4]

[4] L. Bragilevsky and I. V. Bajić, “Tensor completion methods for collaborative intelligence,” *IEEE Access*, vol. 8, pp. 41162–41174, 2020. [4]

[5] A. Dhondea, R. A. Cohen, and I. V. Bajić, “CALTeC: Content-adaptive linear tensor completion for collaborative intelligence,” in *Proc. IEEE ICIP*, (Anchorage, AK), 2021. [4]

[6] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier-Stokes, fluid dynamics, and image and video inpainting,” in *Proc. IEEE/CVF CVPR*, vol. 1, pp. I–355–I–362, 2001. [4]

[7] I. V. Bajić, “Latent space inpainting for loss-resilient collaborative object detection,” in *Proc. IEEE ICC*, 2021. [4]