**Source:** **InterDigital Belgium. LLC**

**Title: [FS\_AI4Media] pCR on intermediate data compression approaches**

**Spec: 3GPP TR 26.927 v0.7.0**

**Agenda item: 9.6**

**Document for: Agreement**

**1. Introduction**

Split operations involve the delivery of intermediate data over the network from one endpoint to another (e.g. UE or a network endpoint). The amount of intermediate data needs to be considered regarding the network capabilities and the service requirements. In particular, S4-240105 mentions the impacts on the uplink as follows: “Main concerns, especially with scenarios in Section 5.1.1.1 in [1], are limited uplink bandwidth and resource sharing at the edge”.

Various compression approaches can be used to reduce the size of intermediate data while still meeting accuracy requirements. This contribution presents the different compression approaches and shows the different characteristics inherent in each of them.

The contribution also adds the related work on MPEG FCM from the PD and introduces text from FCM compression approach.

**2. Reason for Change**

There is no section describing compression approaches or aspects apart from the related work. There is a need to document compression aspects on intermediate data in the TR.

**3. Proposal**

It is proposed to agree and document the following changes to the 3GPP TR 26.927 v0.7.0.

\* \* \* first Change \* \* \* \*

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[aa] 3GPP TR 22.874: "Study on traffic characteristics and performance requirements for AI/ML model transfer".

[bb] Cunningham, P., Cord, M., Delany, S.J. (2008). Supervised Learning. In: Cord, M., Cunningham, P. (eds) Machine Learning Techniques for Multimedia. Cognitive Technologies. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-75171-7_2>

[cc] Supervised Compression for Resource-Constrained Edge Computing Systems https://arxiv.org/pdf/2108.11898.pdf

[ab] AI Model Efficiency Toolkit (AIMET), https://github.com/quic/aimet

[ac] "Application and Verification of NNC in Different Use Cases", MPEG document MDS22894 WG04 N00366, MPEG Video Coding ISO/IEC JTC 1/SC 29/WG 04, July 2023.

[ad] Matsubara, Yoshitomo, Davide Callegaro, Sameer Singh, Marco Levorato, and Francesco Restuccia. "Bottlefit: Learning compressed representations in deep neural networks for effective and efficient split computing." In *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 337-346. IEEE, 2022.

\* \* \* End of first Change \* \* \* \*

\* \* \* Second Change \* \* \* \*

## 4.3 Related work

### 4.3.1 MPEG Feature Compression for Machines (FCM)

Mpeg related work, FCM (Feature Compression for Machines) address features compression as known as intermediate data compression in AI4Media.

A FCM encoder and the FCM decoder would then be part of the intermediate delivery function and intermediate access function, respectively.

Figure 2.2.1-1 illustrates the considered pipeline where, a first part of the Neural-Network-based algorithm is split into two parts. The intermediate features are first encoded on the sender side and embedded in a bitstream, which is decoded at the receiver before inferring the second part of the Neural Network.



Figure 2.2.1-1: FCM pipeline

This standard is expected to be finalized by the end of 2025.

The current baseline considers the use of traditional video compression methods, e.g., H.265/ HEVC or the latest H.266/Versatile Video Coding (VVC) standard, to encode the features that are processed and packed into input frames to the codec. The activity has just started, and new methods are going to be proposed. As the AI models considered in this study rely on Neural Networks, it can be envisioned to optimize the compression of the intermediate features using trained auto-encoders as well, to minimize the size of the bitstreams to be transmitted over 5G, while conserving an acceptable accuracy of the inferred models.

\* \* \* End of second Change \* \* \* \*

\* \* \* Third Change \* \* \* \*

## 6.3 Intermediate data

### 6.3.1 Introduction

Split AI/ML operation is defined as the distribution of AI/ML model inferencing between at least two endpoints, for example a UE and a Network endpoint. The data output from the first endpoint (intermediate data) is delivered to the second endpoint to guarantee the expected user experience on running a particular AI/ML application regarding UE, Network and server capabilities. Requirements for such a split inference service may include avoiding service interruption, and optimizing the network, UE or server resources.

### 6.3.2 Intermediate data size delivery

Intermediate data characteristics depends on various aspects described in clause 4.1 and clause 4.5 including intermediate data volume or size.

Different factors can impact the size of the intermediate data for delivery, which may require the adaptation of split AI/ML operations between the UE and the network:

* AI inference task use-case and requirement: The service requirements on an AI task drive the intermediate data size. For example, a complex AI task for detecting multiple objects in a dense and moving video requires much larger intermediate data than for a simpler AI task on static scene or about a single object.
* AI model for the AI inference task: Different trained AI models for the same AI inference task can be available with different characteristics on not only the AI model architecture and size, but also on the intermediate output size, depending on the split point(s).
* Split point selection: The selection of a split point within an AI model determines the dimension of the intermediate data. The output size at a given split point compared to another may vary from 1 to 5 or more [aa].
* Adapted trained model for split operation: Adapted models can be designed to provide reduced intermediate data at identified split points [cc].
* Optimization: accuracy/quality metrics determine the result of a split inference. Basic precision quantization, from 32 bits to 16/8 bits may reduce the overall size of intermediate data while still meeting the required output result quality/accuracy for the service.
* Inference input video frame rate adjustment: The input frame rate in case of video determines the streaming bitrate of the intermediate data to be delivered. An AI inference task may not produce media content and does not necessarily need to produce an output result at 30 or 60 frames as in the case of video streaming.
* Non-real time delivery: The transmission of intermediate data may not necessarily need to be delivered in a real-time based manner. The result of inferencing a split model on an image, a set of images or a video sequence may not require an immediate result. The transmission of intermediate data can be done progressively with a constrained bandwidth,
* Different input image resolutions may produce different intermediate data size for models with variable input size (e.g. image classification models)

### 6.3.4 Compression related functions

Depending on the AI media service use case (and the required AI task) some compression approaches (e.g., quantization, entropy coding, transformations) can be used to reduce the size of the transferred intermediate data and to adapt the split AI/ML operations between the UE and the network to changing conditions.

Compression functions such as quantization, entropy coding, pruning may be applicable to any intermediate data tensors. It still requires a symmetric decompression function to decode and readapt the intermediate data for inference. This introduces different ratio of reduction of intermediate data size depending on split point configurations. This agnostic compression functions can be available for any model, any split point, any type of model task with different input media data (image, video, audio, text). Agnostic compression evaluation with on-the shelves compression functions provides promising performance in terms of intermediate data size relative to the accuracy of the results.

Compression approaches including an intermediate data bottleneck may be designed to adapt an AI/ML trained model to reduce the amount of intermediate data at least at an identified split point of a model. This requires training for the new bottleneck model. The bottleneck trained model can be selected to replace the original trained model. The compression results can be better, but design is limited to a single split point of a specific model. [ad] shows an example of intermediate data bottlenecks using an embedded autoencoder.

MPEG-FCM clause 4.3 is currently designing compression functions for intermediate data based on existing video compression standards. At the encoder, feature tensors (aka intermediate data) are reduced, converted and mapped onto packed video frames that can be encoded using encoders such as VVC, HEVC, or AVC, e.g., monochrome 10 bits video frames where the tensor channels are spatially packed. The video decoder outputs the packed video frames which are then processed to restore the feature tensors in their original shape, where the conversion, unpacking and feature restoration may use additional metadata transmitted along with the video bitstream. FCM currently addresses AI task for video with very good compression ratios and expect to address agnostic profile in the future.

The table below summarizes the different approaches and characteristics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approaches | Agnostic  | Training required | Number of split points | Reference |
| Bottleneck model | No | Yes | One to several | research [ad] |
| Basic quantization | Yes | No | Any | On the shelves function see Editor’s Note |
| Entropy coding  | Yes | No | Any | On the shelves function see Editor’s Note |
| FC-FCM (current) | No | Yes | One | Editor’s Note |
| FC-FCM (potential/expected) | yes | Yes/No (2 profiles) | Any | Editor’s Note |

Editor’s Note : References should be added.

\* \* \* End of third change \* \* \* \*