**Source: Samsung Electronics Co., Ltd.**

**Title: [FS\_AI4Media] Edits to section on use cases and scenarios**

**Agenda Item: 9.7**

**Document for: Agreement**

# 1 Introduction

This contribution provides minor edits to the section on use cases and scenarios by added the relevant references from TR 22.874.

# 2 Text changes

3 Media-based AI/ML use cases and scenarios

TR 22.874 [1] has identified a set of use cases for AI/ML with the following key operations:

* AI/ML operation splitting between AI/ML endpoints;
* AI/ML model/data distribution and sharing over 5G system;
* Distributed/Federated Learning over 5G system.

These operations have been identified as they require exchange of ML and media data over 5G, and in some cases may have some requirements on the QoS for proper operation.

The use cases and scenarios listed in this technical report, which are described in this clause, are based on a selection of the media-based AI/ML use cases identified in TR 22.874 [1] .

3.1 Object Recognition in Image and Video

Based on clause 5.1 and 5.2 of TR 22.874 [1], this set of use cases, images and video streams are processed to identify and recognize objects and extract some metadata, such as bounding boxes, object labels, movement counters, etc.

The specific scenarios that are considered are the following:

* Delivery of trained ML model(s) for object recognition to the UE in 5GS, including the selection of models for different tasks or environments. This scenario involves the key operation of AI/ML model/data distribution.
* Split inference of trained ML model(s) for object recognition between multiple endpoints, typically between the network and UE. Split points may depend on various factors including UE capabilities, network conditions, and model characteristics. Model characteristics include split inference with a task-specific model head running on the UE for object recognition. For example, in one UE, the task is to recognize pedestrians, whereas in another it is to recognize traffic signs. The core of the network model as well as the input image/video are the same, but the tasks (and their required task-specific models) in the UEs are different. This scenario involves both AI/ML operation splitting, and AI/ML model/data distribution.

Distributed online training of image and video recognition models based on input from different UEs. Depending on the configuration of the ML training framework, different data may need to be delivered between the UEs and the network. Typically a shared model in the network is calibrated continuously based on the training results from all UEs. This scenario involves all the three key operations related to AI/ML model distribution, splitting, and distributed/federated learning.

3.2 Video Quality Enhancement in Streaming

Based on clause 5.3 of TR 22.874 [1], in this use case, the sender and receiver apply parts of an autoencoder DNN model to enhance the quality of a video stream. This is depicted in figure 3.2-1:

说明: A screenshot of a cell phone

Description automatically generated

**Figure 3.2-1**

The sender is typically represented by various media functions in the network, which processes the high-fidelity video using the down-scaling part of a pre-trained DNN model to generate a metadata stream that is streamed together with a lower fidelity encoding of the video. The receiver (UE) runs an inference algorithm (e.g. the up-scaling part of DNN model) on the received metadata and video stream to produce a high-quality video for rendering.

The main scenario in this use case is about streaming intermediate model output data from the network for processing on the UE, involving AI/ML data distribution and operation splitting.

3.3 Crowd-Sourcing Media Capture

This use case and its corresponding scenarios are based on clause 6.2 of TR 22.874 [1]. A set of users attending a live concert and capturing the event on their UEs, use a shared (or a set of shared) DNN model(s) to process and improve their respective captured video and/or audio. Audio and video data may be captured in a noisy environment or an environment with poor lighting conditions. Multiple tasks may then be performed on the processed video and/or audio for media content analysis, e.g. to extract lyrics, annotate the video, improve audio and video quality, translate language, anonymize a face, etc.

This use case involves two different scenarios based on either a device inference or a network inference.

3.3.1 Device inference

The main scenario is to improve the media capture of each UE by using an up-to-date model adapted to the context event.

This scenario may involve the distribution of multiple models to a large number of UEs in a short period of time. The UEs are heterogeneous, running with different types of operating systems (e.g., Android or iOS), supporting different AI/ML engines/frameworks or having different GPU/CPU/NPU and RAM capabilities available for running the AI/ML service on the UE. This will need the distribution of a huge amount of various AI/ML models adapted to the different device capabilities. Depending on each user’s UE, the UE may request the download of a set of DNN models for device inference.

Moving or changing the environment (localization, energy, processing unit, memory, etc.) may need AI/ML model updates, where the DNN models stored in the network may be adapted or updated during the service.

The AI/ML application may optimize the end-to-end latency (e.g., to achieve latency below 1s) or the expected accuracy level of the inference result (e.g., to achieve image recognition precision of 99%) by modifying the model. The desired latency and/or accuracy level can therefore impact the size of the AI/ML model to be distributed. This can be done by:

* optimizing the model accuracy and latency for on-device execution. The model accuracy and execution latency are known, and the optimization may result in bandwidth saving.
* compressing the model for reducing the bandwidth usage and improving the delivery latency. This may affect the accuracy of the model.

If an uncompressed model is sent, accuracy is not affected but delivery latency would depend on the size of the model and the network bandwidth.

The distribution of the AI/ML models for a large number of UEs at the same time may also need to serve the models from different endpoints (e.g., cloud, edge, or other UEs), and may use several or different communication links (e.g. unicast, multicast or broadcast).

3.3.2 Network inference

The main scenario may be the sharing of the input media from multiple sources for network inference, as well as the selection of suitable DNN models according to the UE and/or task.

This scenario requests the UE to upload the media data for network inference. Similarly, to the UE inference, DNN models stored in the network may be adapted or updated during the service for network inferences.

3.4 NLP on Speech

Based on clause 6.3 of TR 22.874 [1], this set of use cases covers a wide range of speech processing use cases, e.g. to perform automatic speech recognition, voice translation, voice commands, speech synthesis, etc.

The AI/ML models for NLP are improved with distributed/federated training using multiple UEs. As more users make use of the service, the quality and accuracy of the models improves. The results of the local training of the models by the UEs are shared with the network.

The main scenario here is about UE downloading a partially trained model identified with its training state for local training, and then sharing the results with the network for distributed/federated learning.

# 3 Proposal

We propose to include the edits in section 2 of this contribution to the relevant clause 3 of the latest version of the Permanent Document.