**Source:** Interdigital Finland Oy

**Title: [FS\_AI4Media]** Use-case 2.3 update

**Agenda Item: 9.8**

**Document for: Agreement**

1. Introduction

This contribution provides an update to the text on use cases currently in the Permanent Document (S4-220500).

2.3 Crowd-Sourcing Media Capture

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.

2.3.1 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 involves the distribution of multiple models to a large number of UEs in a short period of time. The UEs are heterogenous, 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

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).

2.3.2 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 s the UE Similarly, to the UE inference, DNN models stored in the network may be adapted or updated during the service for network inferences.

1. Proposal

We propose to update the text in the PD with the revised text in section 2 of this contribution.