**3GPP TSG-SA4 Meeting SA4#129-e *S4-241440* revision of S4aV240031**

**E-meeting 19 – 23 Aug 2024**

**Source: CMCC, HUAWEI**

**Title: [FS\_AI4Media] pCR on real-time communication scenarios**

**Spec: 3GPP TR 26.927 V****0.8.0**

**Agenda item: 9.6**

**Document for: Discussion and agreement**

**1. Introduction**

The NLP on speech for real-time communication scenarios proposed in S4-231069 was agreed during SA4#124. The sign language translation for real-time communication scenarios proposed in S4aV230085 were agreed during S4-0-e (AH) Video SWG post 126 telco. In SA4#127, S4-240071 was also agreed. This contribution proposes to move the agreed content in clause 4.1.2 and 4.4.1 of the function PD to TR 26.927 V0.8.0

.

**2. Reason for Change**

Move the agreed content in clause 4.1.2 and 4.4.1 of the function PD to TR 26.927

**3. Proposal**

It is proposed to agree the following changes to 3GPP TR 26.927 V0.8.0

\* \* \* 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] Video-Based Sign Language Digit Recognition for the Thai Language: A New Dataset and Method Comparisons. <https://www.scitepress.org/Papers/2023/116437/116437.pdf>

…

[x] <doctype> <#>[ ([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

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

### 4.2.2 Object recognition in image and video

Based on clause 5.1 and 5.2 of TR 22.874 [aa], 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 uses cases are applicable for the different topologies described in clause 5.1, including UE inference only, network inference only and split inferences topologies.

The computationally intensive and memory and power consuming AI/ML inference used to perform this processing requires offloading some inference parts from the mobile device to the edge or a cloud data center.

Split inference of trained ML model(s) for object recognition is distributed between multiple endpoints, typically between the network and UE. Split points may depend on various factors including UE capabilities, network conditions, model characteristics, and user/task specific requirements:

- Device/UE capabilities on running whole or part of model such as the required memory, the processing capabilities, the energy consumption, and the inference latency.

- Network conditions for delivering media and/or the intermediate data. This may include, for example the amount of data to transfer in one shot for an image or at a specific frame rate for video, the required bandwidth in UL and/or DL with different impact on the network load and the related UL and DL network latencies. Network inference latency is also to be considered.

- 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.

- User or task specific requirements. For example, it may be necessary to perform some processing tasks on end-device in order to preserve privacy or because they are delay sensitive operations.

Two main scenarios, both involving either image or video processing are proposed:

a) The UE captures images or video and first feeds the input data to the UE inference model (e.g., to preserve privacy). The UE then uploads intermediate output data from the UE inference model to the network inference, which in turn executes the remaining part of the model (e.g., process the intensive computations) and finally returns the results or a processed image/video to the UE.

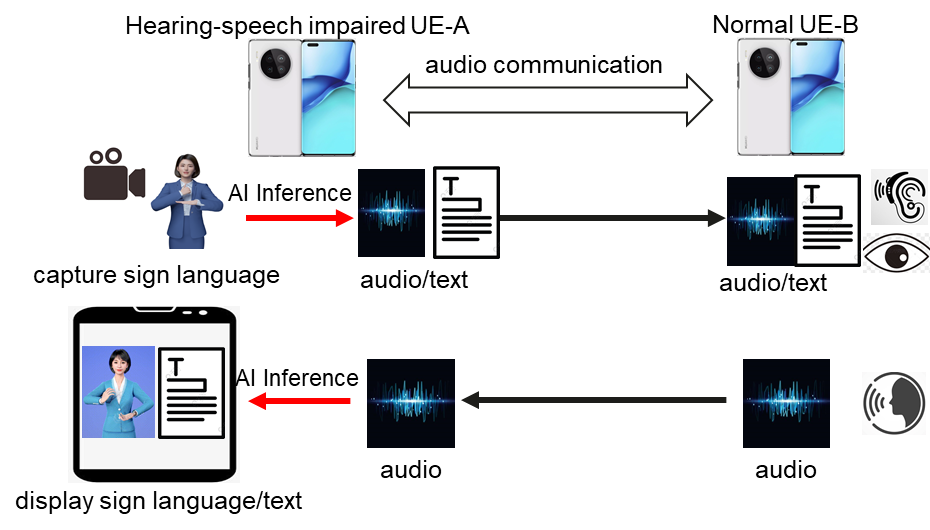
b) Unlike the previous scenario, the UE uploads the captures image or video to the network where a network inference processes inputs video/image, then sends back the intermediate data to the UE inference executing the remaining layers of the model (e.g., task specific operations) and returning the final results.

These scenarios involve the key operation of AI/ML model/data distribution and require the delivery of trained ML model(s) for object recognition to the UE in 5GS, including the selection of models for different tasks or environments and the possible selection of the split points based on the various factors described above

These scenarios also involve the distribution of 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.

#### 4.2.2.1 Scenario: Sign language translation in real-time communication

##### 4.2.2.1.1 Motivation and use case relevance



**Figure 4.2.2.1.1-1: Graphical representation of sign language translation in real-time communication**

Hearing-speech impaired people are unable to have a regular voice call with other people, they can use sign language instead. The sign language can be converted to audio or text in real-time and sent to the normal people. Unimpaired people can still use voice as if they are talking to a normal person, the voice of the normal people can be transferred to an avatar’s sign language or text to display on the screen of the hearing-speech impaired people. This helps hearing-speech impaired people to easily communicate with unimpaired people.

However, sign language AI models typically have several millions of parameters [ad] and may require involvement of network AI inference. For privacy reasons, a hearing-speech impaired user might not want to transmit his/her sign language video stream to the IMS network or the peer user. Therefore, the AI inference for sign language needs to be split between the UE and IMS.

The hearing-speech impaired person UE-A uses a phone to have a voice call with UE-B. UE-A turns on his camera to capture his sign language video, AI inference is performed to translate his sign language to voice or text, the translated voice or text is sent to the UE-B. On the other side, UE-B can use his speaker to talk, the voice of the UE-B can be converted to an avatar’s sign language video stream or text and sent to hearing-speech impaired person UE-A.

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

### 4.2.5 Natural Language Processing (NLP) on speech

Based on clause 6.3 of TR 22.874 [aa], 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.

#### 4.2.5.1 NLP on Speech in real-time communication

NLP on speech in real-time communication can be done on both UE and network, or fully on the network. A use case which is fully completed on network is described as below.

UE-B has subscribed to an intelligent translation service. UE-A initiates an audio/video call and establishes a connection between UE-A and UE-B through the IMS network. When it is detected that UE-B has subscribed to an intelligent translation service, the IMS network serving UE-B decodes the audio stream from UE-A, performs speech recognition and translates it into the required language using AI. Then it sends the translated text along with the video stream or through a separate channel.



**Figure 4.2.5.1-1: workflow for NLP on speech**

\* \* \* End of Changes \* \* \* \*