|  |  |
| --- | --- |
| 3GPP TR 26.927 V0.2.0 (2022-11) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Study on Artificial Intelligence and Machine Learning in 5G media services;  (Release 18) | |
|  | |
|  |  |

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Report is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2021, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

Introduction 5

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 6

3.3 Abbreviations 6

4 Introduction to AI/ML for media 7

4.1 General 7

4.2 Media-based AI/ML use cases and scenarios 7

4.3 Related work 7

5 Media service architecture for AI/ML 7

5.1 General 7

5.2 Architectures and service flows 7

5.2.1 Complete/basic AI/ML model distribution 7

5.2.2 Split AI/ML operation 7

5.2.3 Distributed/federated learning 7

6 Data components for AI/ML-based media services 7

6.1 General 7

6.2 Model data 7

6.3 Intermediate data 7

6.4 Media data 7

6.5 Metadata 8

7 Traffic characteristics 8

7.1 General 8

7.2 Complete/Basic AI/ML model distribution 8

7.3 Split AI/ML operation 8

7.4 Distributed/federated learning 8

8 KPIs 8

8.1 General 8

8.2 List of KPIs 8

9 Potential Normative Work 8

10 Conclusion 8

Annex <A>: <Informative annex title for a Technical Report> 9

Annex <X>: Change history 10

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In drafting the TS/TR, pay particular attention to the use of modal auxiliary verbs! TRs shall not contain any normative provisions.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# Introduction

This clause is optional. If it exists, it shall be the second unnumbered clause.

# 1 Scope

The present document …

# 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: "Vocabulary for 3GPP Specifications".

…

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

# 3 Definitions of terms, symbols and abbreviations

This clause and its three subclauses are mandatory. The contents shall be shown as "void" if the TS/TR does not define any terms, symbols, or abbreviations.

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

Definition format (Normal)

**<defined term>:** <definition>.

**example:** text used to clarify abstract rules by applying them literally.

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

Symbol format (EW)

<symbol> <Explanation>

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

Abbreviation format (EW)

<ABBREVIATION> <Expansion>

# 4 Introduction to AI/ML for media

## 4.1 General

[Editor’s note: Introduction to the concepts of artificial intelligence and machine learning].

## 4.2 Media-based AI/ML use cases and scenarios

### 4.2.1 Introduction

TR 22.874 [aa] 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 [aa].

### 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.

### 4.2.3 Video Quality Enhancement in Streaming

#### 4.2.3.1 Sender-receiver approaches

##### 4.2.3.1.1 End-to-End neural network-based video coding

Based on clause 5.3 of TR 22.874 [1], in this use case, the sender and receiver apply parts of a DNN model (e.g. an autoencoder model) to enhance the quality of a video stream. An example of an autoencoder DNN is depicted in figure 4.2.3-1:

说明: A screenshot of a cell phone

Description automatically generated

Figure 4.2.3-1: Example of DNN-based Down/Up-scaler

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 an intermediate data stream that is streamed together with a lower resolution encoding of the video. The receiver (UE) runs an inference algorithm (e.g. the up-scaling part of DNN model) on using the received intermediate data and video stream to produce a high-quality video for rendering.

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

This use case covers all scenarios where intermediate data stream needs to be sent to the receiver, in addition to a low-resolution video.

##### 4.2.3.1.2 Neural network based post-processing for video coding

A neural network (NN) applies post-processing to a decoded video sequence to enhance the quality of the decoded frames. The post-processing is performed outside the coding loop and does not impact the decoding process of the video. Possible post-processing algorithms include:

- Post-filtering: where the output of the video decoder is provided as input to a NN to improve the quality of the decoded frames. Such improvements include removal of video coding artifacts, subjective quality enhancement, etc.

- Super resolution: where a NN is used to increase the resolution of the output video sequence when the resolution of the display is greater than the resolution of the decoded frames. The use of NN-based approaches in super resolution resampling process increases the quality of the resulting resampled frames.

- NN-based HDR enhancement: a NN is applied for example to enhance a SDR video into an HDR-looking video.

In contrast to 4.2.3.1.1, this approach does not use an intermediate data stream.

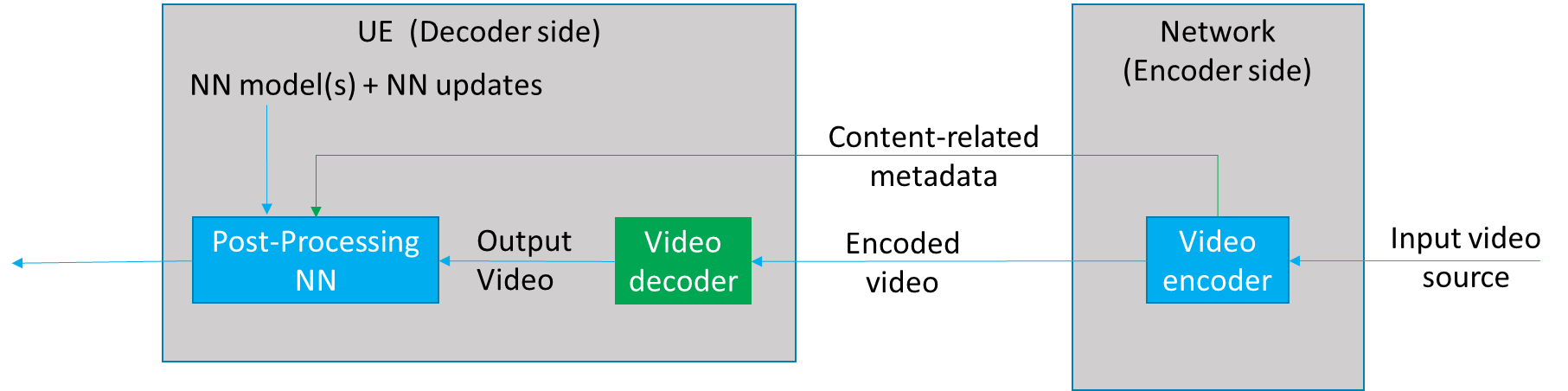


Figure 4.2.3-2: Neural network based post-processing for video coding use-case

Figure 4.2.3-2 depicts a neural-network-based post-processing use-case where pre-trained NN models are used at the receiver to post-process the decoded video to improve the quality. The video encoder processes the input video source to produce and send content-related metadata to the receiver, based on video/image or block, for example. The content-related metadata can be used to select a pre-trained NN model to be applied to a piece of content and to activate or not the selected NN model on it.

### 4.2.4 Crowd-sourcing media capture

#### 4.2.4.1 Introduction

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.

#### 4.2.4.2 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).

#### 4.2.4.3 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.

### 4.2.5 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.

## 4.3 Related work

[Editor’s note: list the AI/ML-related activities in 3GPP and elsewhere, e.g. MPEG…].

# 5 Media service architecture for AI/ML

## 5.1 General

[Editor’s note: Start from basic architectures for the 3 main AI/ML scenarios listed, using 5GMS as a starting point.].

## 5.2 Architectures and service flows

### 5.2.1 Complete/basic AI/ML model distribution

### 5.2.2 Split AI/ML operation

### 5.2.3 Distributed/federated learning

# 6 Data components for AI/ML-based media services

## 6.1 General

[Editor’s note: Identify and document the data types and possible data formats for the different data components listed.].

## 6.2 Model data

## 6.3 Intermediate data

## 6.4 Media data

[Editor’s note: referring to the media data streaming formats and profiles in 26.512.]

## 6.5 Metadata

[Editor’s note: Metadata may include metadata to describe AI/ML model types, metadata for split operation configurations, AI/ML operation endpoint capability metadata etc.]

# 7 Traffic characteristics

## 7.1 General

[Editor’s note: Based on the architectures, identify for the relevant data components for each of the scenarios, the corresponding traffic characteristics (burst size, delay/bandwidth/reliability requirements etc.)]

## 7.2 Complete/Basic AI/ML model distribution

## 7.3 Split AI/ML operation

## 7.4 Distributed/federated learning

# 8 KPIs

## 8.1 General

## 8.2 List of KPIs

[Editor’s note: E.g. Latency, data rate, reliability, accuracy…]

# 9 Potential Normative Work

# 10 Conclusion

Annex <A>:  
<Informative annex title for a Technical Report>

Informative annexes in Technical Reports do not use "(informative") in the title, since all annexes in TRs are informative. Use style "Heading 9" in TRs.

Annex <X>:  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2022-01 | SA4#118e | S4-220498 |  |  |  | Agreements after SA4#118e (S4-220391: TR skeleton) | 0.1.0 |