 ISO/IEC JTC 1/SC 29/WG 04 N0696

**ISO/IEC JTC 1/SC 29/WG 04  
MPEG Video Coding   
Convenorship: CN**

**Document type:** Output Document

**Title:** Application and Verification of NNC in Different Use Cases

**Status:** Approved

**Date of document**: 2025-07-04

**Source:** ISO/IEC JTC 1/SC 29/WG 04

**Expected action:** None

**Action due date:** None

**No. of pages:** 6 (without cover page)

**Email of Convenor:** yul@zju.edu.cn

**Committee URL:** <https://sd.iso.org/documents/ui/#!/browse/iso/iso-iec-jtc-1/iso-iec-jtc-1-sc-29/iso-iec-jtc-1-sc-29-wg-4>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 04 MPEG VIDEO CODING**

**ISO/IEC JTC 1/SC 29/WG 04 N0696**

**July 2025, Daejeon, KR**

|  |  |
| --- | --- |
| **Title** | **Application and Verification of NNC in Different Use Cases** |
| **Source** | **WG 04, MPEG Video Coding** |
| **Status** | **Approved** |
| **Serial Number** | **25474** |

# Introduction

The NNC tools specified in ISO/IEC 15938-17:2024 have been developed to support a range of different model topologies. Due to the need to assess the performance impact of lossy compression in the context of a particular task, a set of tasks from the collected use cases [1] has been selected as benchmark during development of the specification.

In addition, NNC technologies have been tested on other types of models, including some serving the initially identified use cases as well as others. In general, it has been found that NNC can provide good performance also on models with different topologies than those in the benchmark. However, the set of coding tools to be applied may need to be adjusted, as some may provide less or no coding gains for some layer types.

This document summarises in which use cases (those selected for the evaluation framework [2] during development, as well as others, covering a variety of topologies and layer types) the NNC technologies have been verified so far, and describes how an experiment for testing on new tasks can be set up. It also provides a template for collecting information about new tasks. Such new tasks are expected to use networks different from those used in the experiments during specification developments. Examples of such tasks are AI-based point cloud compression, scene representation using NERFs and tasks using (vision) transformers. Additionally, NNC has been applied to (intermediate) data generated by neural networks, incremental neural updates, e.g. in Federated Learning, and general tensors in AI-based media, such as function coefficients.

# Use cases reported

## Image/video processing and coding

**Image super-resolution** has been verified with both the EDSR and Swin2SR networks. The experiments perform 2x and 4x super-resolution on the DIV2K dataset, using PSNR, SSIM and LPIPS as metrics.

NNCodec is used with dependent scalar quantization, not activating further tools and reporting results for QP values between -2 and -46.

Additional details about the model used and the experiment setup can be found in m64169 (Jul. 2023) for Swin2SR and m62646 (Apr. 2023) for EDSR.

**Image restoration** using the NAFNet model has been verified on the GoPro dataset using PSNR as the metric. Additional details about the model used and the experiment setup can be found in m62646 (Apr. 2023).

A **learned quality metric** (LPIPS) that uses a NN backbone for feature extraction has been verified. The experiments perform image quality evaluation on DIV2K dataset upscaled 2x and 4x through bilinear, bicubic, and Swin2SR, and compares the scores of the quality metric determined with the uncompressed and compressed feature extractors. NNCodec is used with dependent scalar quantization, not activating further tools and reporting results for QP values between -2 and -46. Additional details about the model used and the experiment setup can be found in m64447 (Jul. 2023).

End-to-end **image compression** using an autoencoder has been evaluated as part of the evaluation framework of NNC. CIFAR100 has been used as dataset, and the reconstructed images have been assessed using PSNR and SSIM. Additional details about the model used and the experiment setup can be found in [2].

## 3D processing and coding

**Implicit neural representations** such as NERFs are promising 3D scene representations, which represent a scene as parameters of a neural network trained on the scene. The 3D dynamic NeRF models DyNERF and MixVoxels have been verified on the CBABasketball and Mirror sequences, using PSNR as the metric. Additional details about the model used and the experiment setup can be found in m63165 (Apr. 2023).

**Point cloud compression** has been verified on the GRASP-Net network, and for evaluation D1 PSNR and D2 PSNR, which are point cloud spatial distortion assessment metrics, were used. Additional details about the model used and the experiment setup can be found in m64314 (Jul. 2023).

## Classification, detection and segmentation

As part of the evaluation framework of NNC, **image classification** on the ImageNet dataset has been evaluated, using top-1 and top-5 accuracy as metrics. Convolutional models with different size, complexity and layer types, in particular VGG16, ResNet50 and MobileNet v2 have been assessed. Additional details about the model used and the experiment setup can be found in [2].

Recently, transformers have become the state of the art architecture for a number of vision tasks. **Image classification** using SWIN Transformer has been evaluated on ImageNet1K, using top-1 accuracy as a metric. Additional details about the model used and the experiment setup can be found in m62646 (Apr. 2023).

The above mentioned and related models (ResNet variants, MobileNet v2, ViT Vision Transformers) were also tested in Federated Learning scenarios. More precisely, training 16 client devices for 120 communication rounds on CIFAR-100 and ImageNet-200 **image classification** tasks from scratch, and transfer learning 16 clients from ImageNet-1k to Pascal VOC for 60 communication rounds. Coding results and related references can be found in [3].

**Image classification** hasalso been examined in a *SplitFed* Learning scenario, where a global model is collaboratively trained across multiple instances, with the model architecture split between clients and server. Clients transmit intermediate activations (“smashed data”) from their local “cut layers” to the server-side portion of the model – typically larger and computationally more complex – which computes gradients with respect to the smashed data. These gradients are then sent back to the clients for backpropagation through their local layers. This use case requires coding multiple data modalities (i.e., activations and gradients). Detailed results and experimental configuration are provided in document m72294 (April 2025).

**Object detection** has been verified on a Yolo v3 network, using F1 metric in the evaluation. Additional details about the model used and the experiment setup can be found in m51436 (Oct. 2019, using a preliminary version of the specification).

SWIN Transformer has also been assessed for **object detection** on the MS COCO dataset, using box mAP as metric. Additional details about the model used and the experiment setup can be found in m62646 (Apr. 2023).

## Acoustic scene classification

As part of the evaluation framework of NNC, the DCASE benchmark's 2017 task 1 acoustic scene classification dataset and the 2019 baseline model have been used, with classification accuracy as the metric. Additional details about the model used and the experiment setup can be found in [2].

## Recommender system

Two different stages of a recommender system for user views have been verified, using top100 accuracy metric: The candidate generation neural network contains an embedding layer (for movie watches) and three fully connected layers. Overall, the candidate generation model has ~18M trainable parameters. The ranking neural network contains an embedding layer with sub-embeddings for users, movies chosen genres, as well as one fully connected layer. Overall, the ranking model has ~12M trainable parameters.

Both stages were evaluated in central as well as federated leaning scenario.

Additional details about the model used and the experiment setup can be found in m62636 (Apr. 2023).

## Adaptive bitrate selection using reinforcement learning

For adaptive streaming, reinforcement learning of adaptive bitrate models using Pensieve has been verified. The experiment has been done compressing also incremental updates in five rounds, using the average reward in reinforcement learning as metric. Additional details about the model used and the experiment setup can be found in m58995 (Jan. 2022).

## NLP

NLP has been verified on a BERT network, using F1 metric in the evaluation. Additional details about the model used and the experiment setup can be found in m51436 (Oct. 2019, using a preliminary version of the specification).

## LLMs

The application to large language models has been verified by testing with Llama2 on different common sense reasoning datasets (measuring average accuracy) and perplexity on WikiText-2. Additional details about the experiment setup and results can be found in m66483 (Jan. 2024).

# Networks used in NNC verification

The following table reports the layer types or named model topology (where applicable). The compression rate is the size of the resulting NNC bitstream, expressed as percentage of the original size of the neural data at a working point that yields approximately the same performance as the uncompressed variant in terms of a metric relevant to the task (e.g., top-k accuracy for classification, PSNR or SSIM for compression or super-resolution). Where a range is specified, this indicates performances reported for different datasets or network variants. The documents reporting the results provide in most cases a number of additional results for other working points

| **Application** | **Layer types / model** | **Compression rate at transparent performance** |
| --- | --- | --- |
| Image super-resolution | Vision transformers (SWINv2) | 9-15% |
| Image super-resolution | 2D convolutions (EDSR) | 15% |
| Image restoration | 2D convolutions (NAFNet) | 18% |
| Learned quality metric (LPIPS) | 2D convolutions (AlexNet backbone), fully connected | 9% |
| Image/Video Compression | 2D convolutions | 17% |
| Implicit Neural Video Representation (e.g. NERF) | DyNERF, MixVoxels | 10-20% |
| Point cloud compression | 3D convolutions (GRASP-Net) | 20% |
| Visual object classification | 2D convolutions, pooling, batch-normalisation, fully connected (VGG16, ResNet50, MobileNet v2) | 3-12% |
| Visual object classification | Vision transformers (SWIN) | 10-12% |
| Visual object classification in Federated Learning (from scratch and transfer) | Differential model / layer updates of ResNet variants, MobileNet v2 and Vit-B/16 | 2-3% (from scratch) 0.1-0.6% (transfer) |
| Visual object classification in a SplitFed setting | Activations and gradients from/to a ResNet56 model split after its first residual block and a MobileNet v2 model split after its second block | 3-5% |
| Object detection | Vision transformers (SWIN) | 16% |
| Object detection | Yolo v3 (2D convolutions, pooling, batch-normalisation, fully connected) | 10% |
| Acoustic scene classification | convolutions, fully connected | 4% |
| Recommender system | feature embedding, fully connected; also tested in federated learning setting | 2-4% |
| Adaptive bitrate selection using reinforcement learning | convolutions, fully connected (Pensieve) | 20% |
| NLP | transformer encoders (BERT) | 15% |
| Commonsense reasoning (text generation) | Large language models (Llama2 7B) | 17% |

The verification of NNC in a diverse range of use cases, covering networks with different topologies and layer types, has shown that a compression rate of 10-15% can be achieved for many models. Some models can be compressed even down to 2%, while models that were already designed for compactness can be compressed to about 20%. In the context of federated learning, compressing differential updates yields compression rates between 0.1-3%, attributable to the sparser representations of model differences relative to a shared base model. In SplitFed Learning, intermediate activations and gradient data can be compressed to approximately 3-5% of their original size on average. However, this ratio varies depending on the training stage (e.g., in later stages of training, gradients tend to be smaller and often more compressible). These numbers refer to compression without performance loss in terms of the respective metric of the application. In many of the applications, small performance reductions can be traded off against further reducing the model size. Together with the properties of the NNC technology these results show that the NNC standard fulfils the requirements defined in [1].

# Evaluating NNC in a new task

In order to assess the performance in a new task, it is required to measure both the compression performance as well as the performance of using the weights compressed and reconstructed network in the task. The compression performance is measured in terms of the ratio of the encoded bitstream and the original model, and if relevant, in terms of the ratio memory footprint of the reconstructed model and the original model. The performance of the network in the task is measured using a task-specific metric, such as classification accuracy or PSNR of an encoded image.

In order to perform experiments of in a new task, the following minimal set must be available:

* Trained neural network for the task
* Validation data set
* Performance metric(s)

However, as some coding tools benefit strongly from fine-tuning, at least a partial training dataset is required to fully benefit from some tools. For this purpose, also training parameterisation to be used for fine-tuning should be provided.

A basic workflow for an experiment consists of the following steps:

* Train the source model for the task
* Determine task-specific performance metrics
* Apply compression using NNC (obtaining a compressed bitstream and reconstructed model)
* Measure size of bitstreams
* Determine task-specific performance metrics, using the compressed and reconstructed model

To simplify running experiments, the compression experiment can be restrict to tools that do not alter the topology of the network (such as tensor decomposition). In such a case the set of

reconstructed weights can be plugged into the original pipeline. Otherwise some code adjustments may be necessary to load the adjusted model topology.

The most recent version of the NNC reference software obtained from [4]. An open source implementation of ISO/IEC 15938-17 is also available [5]. In cases where fine-tuning shall be applied, the training framework and (a subset of) the training data need to be available during compression.

The evaluation framework for NNC [2] provides further details about the evaluation metrics.

We welcome reports about the use of NNC in new applications, providing the following information:

* Brief description of use case/application
* Information about the network: network topology (including description of any non-standard layers), URL to obtain model and (if already available) trained weights
* Description of evaluation environment in which experiments have been performed (e.g., dataset, evaluation metric(s))

# References

[1] Use cases and requirements for neural network compression for multimedia content description and analysis, WG11 N18770, Oct. 2019

[2] Evaluation Framework of Compression of Neural Networks for Multimedia Content Description and Analysis, WG11 N18575, Jul 2019

[3] ISO/IEC JTC 1/SC 29/AG 03 N139, White Paper on Neural Network Coding, Online, Jan. 2024

[4] NN Compression Test Model (NTCM), <http://mpegx.int-evry.fr/software/MPEG/NNCoding/NCTM>

[5] D. Becking et al., "NNCodec: An Open Source Software Implementation of the Neural Network Coding ISO/IEC Standard," ICML Workshop Neural Compression 2023. <https://github.com/fraunhoferhhi/nncodec>