Moving Picture, Audio and Data Coding
by Artificial Intelligence
www.mpai.community

MPAI Technical Specification

AI-Enhanced Video Coding (MPAI-EVC) – Up-sampling Filter for Video Applications EVC-UFV

V1.0

WARNING

Use of the technologies described in this Technical Specification may infringe patents, copyrights or intellectual property rights of MPAI Members or non-members.

MPAI and its Members accept no responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this Technical Specification.

Readers are invited to review Notices and Disclaimers.

AI-Enhanced Video Coding (MPAI-EVC) – Up-sampling Filter for Video Applications EVC-UFV

1 Foreword

The international, unaffiliated, non-profit ***Moving Picture, Audio, and Data Coding by Artificial Intelligence (MPAI)*** organisation was established in September 2020 in the context of:

1. **Increasing** use of Artificial Intelligence (AI) technologies applied to a broad range of domains affecting millions of people
2. **Marginal** reliance on standards in the development of those AI applications
3. **Unprecedented** impact exerted by standards on the digital media industry affecting billions of people

believing that AI-based data coding standards will have a similar positive impact on the Information and Communication Technology industry.

The design principles of the MPAI organisation as established by the MPAI Statutes are the development of AI-based Data Coding standards in pursuit of the following policies:

1. Publish upfront clear Intellectual Property Rights licensing frameworks.
2. Adhere to a rigorous standard development process.
3. Be friendly to the AI context but, to the extent possible, remain agnostic to the technology thus allowing developers freedom in the selection of the more appropriate – AI or Data Processing – technologies for their needs.
4. Be attractive to different industries, end users, and regulators.
5. Address five standardisation areas:

   1. *Data Type*, a particular type of Data, e.g., Audio, Visual, Object, Scenes, and Descriptors with as clear semantics as possible.
   2. *Qualifier*, specialised Metadata conveying information on Sub-Types, Formats, and Attributes of a Data Type.
   3. *AI Module* (AIM), processing elements with identified functions and input/output Data Types.
   4. *AI Workflow* (AIW), MPAI-specified configurations of AIMs with identified functions and input/output Data Types.
   5. *AI Framework* (AIF), an environment enabling dynamic configuration, initialisation, execution, and control of AIWs.
6. Provide appropriate Governance of the ecosystem created by MPAI Technical Specifications enabling users to:
   1. *Operate* Reference Software Implementations of MPAI Technical Specifications provided together with Reference Software Specifications
   2. *Test* the conformance of an implementation with a Technical Specification using the Conformance Testing Specification.
   3. *Assess* the performance of an implementation of a Technical Specification using the Performance Assessment Specification.
   4. *Obtain* conforming implementations possibly with a performance assessment report from a trusted source through the MPAI Store.

Today, the MPAI organisation rests on four solid pillars:

1. The MPAI Patent Policy specifies the MPAI standard development process and the Framework Licence development guidelines.
2. *Technical Specification: Artificial Intelligence Framework (MPAI-AIF)* specifies an environment enabling initialisation, dynamic configuration, and control of AIWs in the standard AI Framework environment depicted in Figure 1. An AI Framework can execute AI applications called AI Workflows (AIW). An AIW includes interconnected AI Modules (AIM). MPAI-AIF supports small- and large-scale high-performance components and promotes solutions with improved explainability.

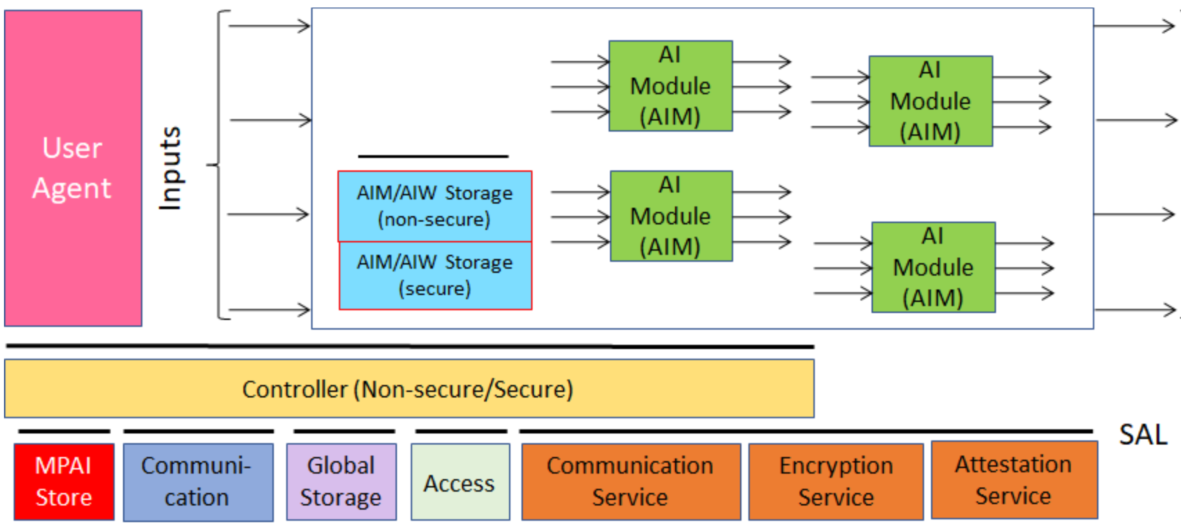

Figure 1 – The AI Framework (MPAI-AIF) V2 Reference Model

3. ***Technical Specification: Data Types, Formats, and Attributes (MPAI-TFA) V1.0*** specifies Qualifiers, a type of metadata supporting the operation of AIMs receiving data from other AIMs. Qualifiers convey information on Sub-Types (e.g., the type of colour), Formats (e.g., the type of compression and transport), and Attributes (e.g., semantic information in the

Content). Although Qualifiers are human-readable, they are only intended to be used by AIMs. Therefore, Text, Speech, Audio, and Visual Data exchanged by AIWs and AIMs should be interpreted as being composed of Content (Text, Speech, Audio, and Visual as appropriate) and associated Qualifiers. The specifications of most MPAI Data Types reflect this point.

4. *Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V1.1* defines the following elements:
5. Standards, i.e., the ensemble of Technical Specifications, Reference Software, Conformance Testing, and Performance Assessment.
6. Developers of MPAI-specified AIMs and Integrators of MPAI-specified AIWS (Implementers).
7. MPAI Store in charge of making AIMs and AIWs submitted by Implementers available to Integrators and End Users.
8. Performance Assessors, independent entities assessing the performance of implementations in terms of Reliability, Replicability, Robustness, and Fairness.
9. End Users.

The interaction between and among actors of the MPAI Ecosystem are depicted in Figure 2.

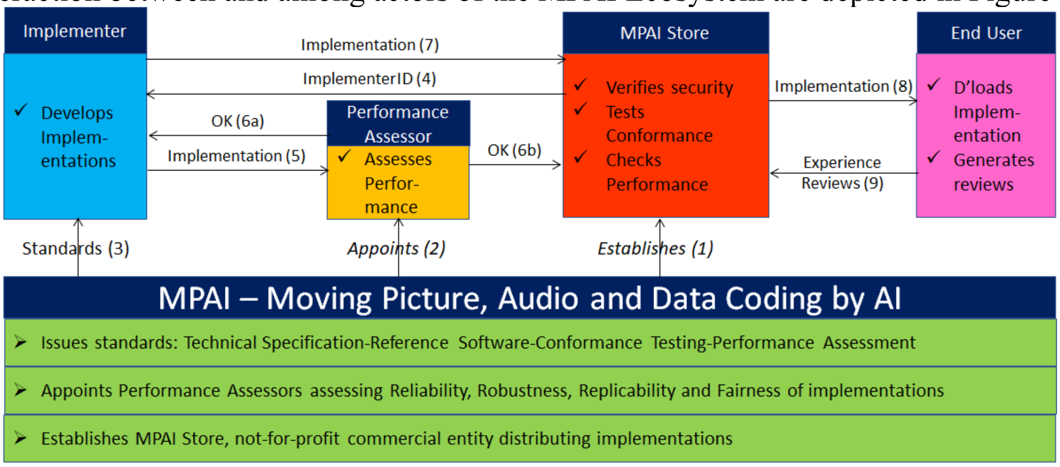

Figure 2 – The MPAI Ecosystem

2 Introduction (Informative)

Many applications require video filtering, for instance in video compression where the full data rate reduction process involves capture, down-sampling, encoding, transport, decoding, up-sampling, and rendering.

Typical up-sampling technologies used in video coding applications are Bi-cubic and Lanczos filters. However, they have a major limitations because they perform purely mathematical operations on the decoded video sequence.

Up-sampling filters resulting from a process of training using a large number of video sequences are highly non-linear and are able to reproduce the original-resolution video sequences beyond the simple interpolation of pixels in the sub-sampled video frames.

Technical Specification: AI-Enhanced Video Coding (MPAI-EVC) – Up-sampling Filter for Video applications (EVC-UFV) V1.0 specifies procedures to design up-sampling filter for video applications based on Super Resolution and to reduce their complexity..

3 Scope

Technical Specification: AI-Enhanced Video Coding (MPAI-EVC) – Up-sampling Filter for Video applications (EVC-UFV) V1.0 – in the following also called as EVC-UFV V1.0 or simply EVC-UFV –

1. Specifies
    1. A procedure to design up-sampling filters for video applications based on super resolution techniques
    2. A procedure to reduce the complexity of the designed up-sampling filter to a specified level of complexity.
2. Provides the parameters of two specific complexity-reduced up-sampling filters for
    1. Standard definition to high definition
    2. High definition to ultra high definition.

Users are informed that the high definition to ultra high definition up-sampling filter can be used for standard definition to high definition as well with limited performance loss.
The Terms used by EVC-UFV are defined in Table 1. All MPAI Terms are accessible online.
EVC-UFV V1.0 has been developed by the AI-Enhanced Video Coding group of the Requirements Standing Committee. MPAI may decide to develop new versions extending EVC-UFV V1.0 or develop new specifications in the same of related areas.

4 Definitions

Capitalised Terms in EVC-UFV V1.0 have the meaning defined in Table 1.
A dash "-" preceding a Term in Table 1 indicates the following readings according to the font:
1. Normal font: the Term in the table without a dash and preceding the one with a dash should be read after that Term. For example, "Risk" and "- Assessment" will yield "Risk Assessment".
2. *Italic* font: the Term in the table without a dash and preceding the one with a dash should be read before that Term. For example, "Descriptor" and "- *Financial*" will yield "Financial Descriptors."

All MPAI-specified Terms are defined online.

Table 1 – Terms defined and/or used by EVC-UFV

Term	Definition
Activation Function	A mathematical function determining whether a neuron should be activated based on the input to the neuron.
Block	A fundamental component or module within a neural network architecture.
Channel	A single slice of data along the depth of the tensor. For example, in an image, depth is a single channel of the colour space.
Data Augmentation	A technique that increasing the training dataset with new training examples obtained by altering some features of the original training dataset.
Densely Residual Laplacian	
– Module	(DRLM) A set of RUs where each RU is followed by a Concatenation Layer.
– Network	A Deep Learning Model that combines dense connections, residual learning, and Laplacian pyramids to enhance image restoration tasks like super-resolution and denoising.
Dilation	A technique for expanding a convolutional kernel by inserting holes or gaps between its elements.

Dependency Graph	(DepGraph) a framework to simplify the Structured Pruning operation of neural networks.
Epoch	The total number of iterations of all the Training data in one cycle for training a Machine Learning Model.
Feature maps	The outputs of convolutional layers.
Fine Tuning	The Process of re-training a model trained on a dataset A on a new dataset B.
Importance	The arithmetic mean of all Parameters of the Channel.
Inference	The process of running a Model on an input to produce an output.
Initial Number of Parameters	The number of parameters of the unpruned Model.
Input/Output Decomposition	The process of breaking down complex input data into simpler, more manageable components or features, and then using these to generate meaningful outputs.
Dependency Graph	A graph representing the dependency between any input and output decomposition.
Laplacian	
– Attention Unit	(LC) A set of Convolutional Layers with a square filter size and Dilation that is greater than or equal the filter size.
– Pyramid	A representation of an image that uses the difference between the application of a Gaussian Filter and the image at different resolution values.
Layer	A set of parameters at a particular depth in Neural Network.
– Concatenation	The process of combining multiple layers into a single tensor.
– *Convolutional*	A Layer of Neural Network Model that applies a convolutional filter over the input.
Learning	
– *Deep*	A type of Machine Learning that uses artificial Neural Networks with many Layers to learn patterns from data.
– *Machine*	A class of algorithms that enable computers to learn from data thus enabling them to make predictions called inferences from new data.
– *Rate*	A value linked to the step size at each iteration toward a minimum of the Loss Function.
– *Sparsity*	Learning strategy to detect the most relevant features of a Model in the set of all the Model features for a particular learning task.
Loss function	A mathematical function that measures the distance between the output of a Machine Learning Model and the actual value.
Maximum Pruning Ratio	The highest percentage of a neural network’s parameters (weights, neurons, or connections) that can be removed without causing a significant performance drop.
Model	
– *Deep Learning*	An algorithm that is implemented with a multi-Layered Neural Network.

– *Machine Learning*	An algorithm able to identify patterns or make predictions on datasets not experienced before.
– *Pre-trained*	A Model that has been trained on a Dataset possibly of a different from the one in which the Model has to be used.
– Recovery Phase	A training procedure applied after Pruning to recover part of the performance lost because of a Pruning Algorithm was applied.
Neural Network	(Also Artificial Neural Network), A set of interconnected data processing nodes whose input and output connections are affected by Weights.
Neuron	A data processing node in a Neural Network.
Patch	A squared subset of a frame, whose size if often multiple of 2, used to define the square size (e.g., 8×8, 16×16, 32×32).
Parameter	The multiplier of the input to a Neural Network neuron learned via Training.
Performance Criterion	The percentage ratio of the Pruned Model and the unpruned Model that is considered acceptable.
Pre-training	A phase of Neural Network Model Training where a model is trained on an often-generic dataset to allow it to learn a more generic representation of the task.
Pruning	The process of removing less important parameters (like weights or neurons) from a neural network to reduce its size and computational requirements, while retaining the model performance.
– Group	A group of decompositions that include those dependent on each other and subject to joint pruning.
– Target	The percentage of the Model parameters – computed with reference to to the Initial Number of Parameters – be be pruned.
– *Growing Regularisation*	A technique that sets the unimportant weights to zero before eventually removing them.
– *Learning-Based*	A set of Pruning techniques which require variations of learning in order to be implemented.
– Recovery	A method that involves retraining a pruned neural network to regain any lost accuracy.
– *Structured*	A method that removes entire components like neurons, filters, or channels, resulting in a smaller dense model architecture.
– *Unstructured*	Unstructured Pruning focuses on removing single redundant neurons. However, creating a sparse model representation which does not compute faster in common hardware.
Rectified Linear Unit	(ReLU) An Activation Function whose output is the input if it is positive and zero otherwise.
Residual	
– Block	A Block composed of concatenated DRLM modules where each module is followed by a Concatenation and Convolutional Layer.

– Function	A function that provides the difference between the input and the desired output of a layer or a stack of layers.
– Neural Network	(ResNet) A Neural Network whose Layers learn Residual Functions with reference to the inputs to each Layer.
– Unit	(RU) A set of alternate ReLU and Convolutional Layer.
Resolution	
– *Visual*	The dimension in pixels, expressed as width × height (e.g., 1920×1080), indicating how many pixels make up an image or a video frame.
Saliency Value	A value representing the ability of an image or a video frame to grab the attention of a human.
Sampling	
– *Down-*	The process of reducing the Visual Resolution.
– *Up-*	The process of increasing the Visual Resolution.
Super Resolution	The technique enabling the generation of High-Resolution Visual Data from a low-Resolution one.
Training	The process of letting a Model experience examples of inputs that the Trained Model might experience or outputs that the Trained Model should produce, or both.
– Set	The dataset used to train a Model.
Validation	The process of evaluating a Trained Model on a dataset (called Validation Set) that the Model has not experienced during training.
– Score	The error of a Model on the Validation Set.
– Set	The data set used to check the performance of a Model to know when to stop the Training.
Video Frame	An image drawn from for the sequences of images composing a video.

5 References

5.1 Informative References

1. A. Jamil and A. Artusi, "Computational efficient deep learning-based super resolution approach", Proc. SPIE 12571, Real-time Processing of Image, Depth and Video Information 2023; DOI: 10.1117/12.2665645
2. Diederik P. Kingma, Jimmy Ba; Adam: A Method for Stochastic Optimization; 3rd International Conference for Learning Representations, San Diego, 2015.
3. Saeed Anwar, Nick Barnes, "Densely Residual Laplacian Super-Resolution", IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2020.
4. Y. He, X. Zahang and J. Sun; Channel Pruning for Accelerating Very DeepNeural Network.
5. Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, Xinchao Wang; DeepGraph: Towards any structural Pruning Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 16091-16101

6 EVC-UFV V1.0 Design Procedure

6.1 Introduction

This chapter specifies the steps enabling a User to design a neural network able to up-sample a video sequence to a higher resolution than the current video resolution through the following steps:

1. Selection of video sequences for use in the development of the Training Dataset.
2. Creation of the Training Dataset.
3. Pre-Training phase.
4. Fine-Tuning phase.
5. Definition of the Up-sampling Network Model.

This chapter also provides Reference Software for training a Neural Network using the procedure specified in this chapter.

6.2 Data Preparation

Assuming the the target resolution of m rows by n columns, the training dataset of frames to be used in the training process, consists pairs of input frames of resolution m/2 by n/2 and output frames of resolution m by n. If the input frames are not available, they may be obtained from the output frames by using a down-sampling filter.
To reduce the computing time required for training, as well as to overcome memory management issues, patches extracted from the input and output frames may be used. The resolutions of the patches are h/2 by k/2 and h by k for the input and output patches, respectively. The number of patches extracted from the frames shall be an appropriate smaller number than the total number of patched in the frame and h and k shall be appropriately smaller than m and n, respectively.
Patches may be extracted with different methods, e.g., randomly, feature-based etc.
To ensure that the trained filter is applicable to to a wide range of video material outside of those used for training, Augmentation maybe used. The size of the training dataset is increased by transforming patches or frames, e.g., by rotating, adding noise, mirroring, etc.

6.3 Pre-Training

Although the model can be trained starting from an untrained or from a trained model, the latter provides better result by fine tuning a model that was pre-trained using the method specified below.
The pre-training method is performed with the following process:

1. The pre-training set shall have a size of 800 high definition images at least.
2. The images are diversifies through data Augmentation with the following process:
    1. Selection of square patches.
    2. Each patch is randomly changed by applying one of more of the following:
        1. Rotations by multiples of 90°.
        2. Horizontal flipping.
        3. Vertical flipping.
3. The pre-training uses the following:
    1. Batch size of 4.
    2. Backpropagation algorithm according to ADAM with default parameters of β1 =0.9, β2 =0.999, and ϵ= 10−8.
    3. The learning rate is fixed to 10^{-4} originally and then decreased to half after every 2^4 iterations.

6.4 Fine-Tuning

The fine-tuning is performed with the following process:

1. Select a fine-tuning dataset data for the specific application domain, e.g., in case of video application, encoded and decoded video sequences
2. Compute the Saliency Value.
3. Retain the patch if it is adequately separated in the Cumulative Distribution Function of the Saliency Value.
4. Augment the dataset size by randomly changing the patch by applying one of more of the following:
    1. Rotations by multiples of 90°.
    2. Horizontal flipping.
    3. Vertical flipping.
5. The first four DLRM Residual Blocks are frozen while the remaining DRLM are trained.
6. The fine tuning is applied for 200 epochs using a batch size of 4.
7. The learning rate is initially set to 10^{-5} and then reduced during learning with a ReduceLROnPlateau scheduler with Patience 15 and learning rate factor of 0.5.
8. The ADAM optimization is used with initial parameters 0.9, 0.999, 10-8 for β1, β2, and ϵ respectively.
9. The extracted pair of patches for the training set have a size of 64×64 pixels for the input and 128×128 pixels (or the output (2x up-sampling).
10. The data sets is split into training and validation sets with a 20% validation dataset.

The reference implementation of the training process will be made available at the MPAI Git.

6.5 Development of the Up-sampling Network Model

The starting point is the Densely Residual Laplacian Super-Resolution Network depicted in Figure 1.

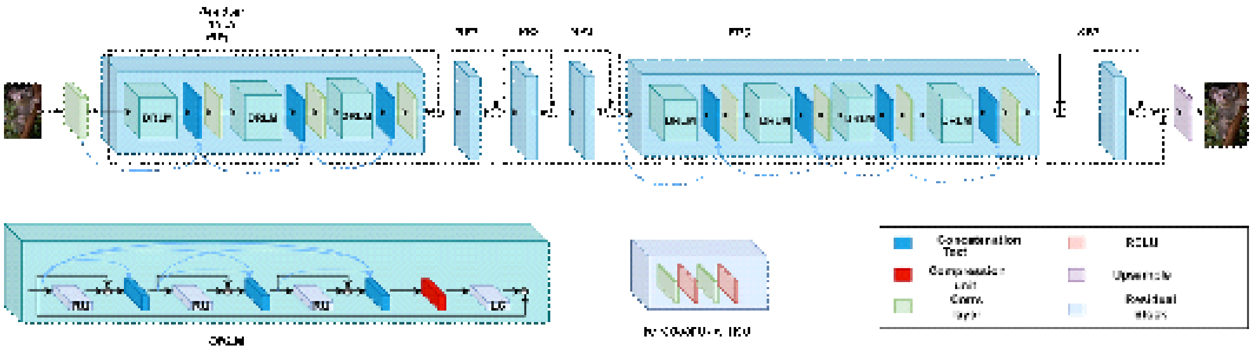

Figure 1 – Densely Residual Laplacian Super-Resolution Network (DRLN).

As shown in Figure 1, the main component of the DRLN architecture is a Residual Block which is composed of the Densely Residual Laplacian Modules (DRLM) and a convolutional layer. Each DRLM contains three Residual Units, one compression unit, one Laplacian attention unit with Dilation that is greater than or equal to the filter size. Each Residual Unit consists of two convolutional layers and two ReLU Layers. All DRLM modules in each Residual Block and all Residual Units in each DRLM are densely connected. The Laplacian attention unit consists of three convolutional layers with filter size 3×3 and dilation equal to 3, 5, 7. All convolutional layers in the network, except the Laplacian one, have filter size 3×3 with Dilation equal to 1. Throughout the network, the number of Feature Maps is 64.

Figure 2 gives the structure of the network having the following characteristics:

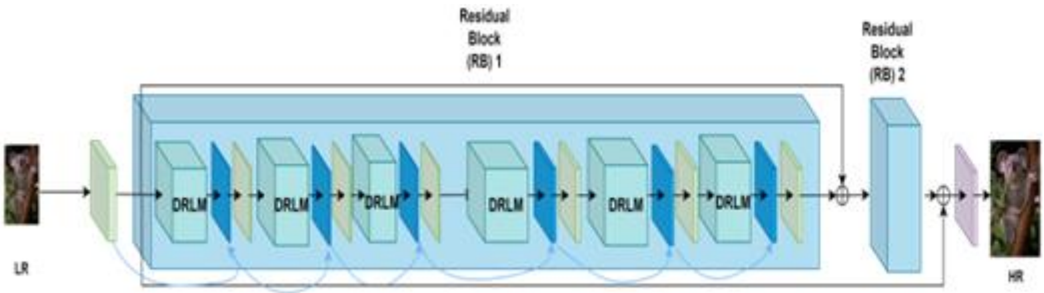

Figure 2 – Structure of the EVC-UFV Up-sampling Filter

Table 1 compares the values of the original and simplified Neural Network Model.

	Original	**Simplified**
Residual Blocks	6	2
DRLMs per Residual Block	3	6
Residual Unit per DRLM	3	3
Hidden Convolutional Layers per Residual Unit	2	1
Input Feature Maps	64	32

7 6 Reference Software

The Reference Software is released as Open Source Software with BSD 3-Clause Licence and can be downloaded from the MPAI Git (registration required).

8 EVC-UFV V1.0 Complexity Reduction

8.1 Introduction

This chapter specifies The steps required to implement the pruning method that enables a User to produce an Up-sampling network with reduced complexity, starting from a fine-tuned model produced using the procedure specified in Design Procedure.

The Pruning Algorithm specified below in natural language and pseudo-code is applied to the Fine-Tuned or Pre-trained Model (in the following "Model"). This chapter also includes the Reference Software of the Pruning Algorithm.

The Pruning Algorithm also requires a Pruning Dataset; this can be the one used in the Design Procedure.

In the following the Pruning Algorithm is specified in natural language and in pseudo-python code.

8.2 Pruning Algorithm in natural language

1. Set the value of the target Performance Criterion.
2. Set the Pruning Step Size as the percentage of parameters to be removed at each pruning iteration
3. Compute the Dependency Graph to obtain the Pruning Groups.
4. Starting from the non-pruned network, execute a set of Steps numbered from 1 to N (maximum number of Steps) until the Pruned Model satisfies the Performance Criterion or exceeds the Maximum Pruning Ratio:
    1. Set the Iteration Pruning Target by adding the Pruning Step Size.
    2. Apply Sparsity Learning to the Model.
    3. Evaluate the Importance of each Channel in each Layer of the Model.
    4. Remove the Channels until the Pruning Target is met, starting from the one with the lowest Importance.
    5. For a predefined number of epochs E:

        1. Train the pruned Model over the training dataset for 1 epoch.
        2. Evaluate the MSE of the retrained model.
        3. If the MSE is the smallest of those achieved so far, save the current model.
    6. Select the Model with the smallest MSE.
5. Save the current Pruned Model.

8.3 Pruning Algorithm in pseudo code

```
# dependecy graph computation
graph = dependecy_graph(MODEL)
original_error = calc_error_mse(MODEL, DATASET.val)
current_pruning_target = PRUNING_STEP_SIZE

# starting pruning process
while (model_accuracy > original_error * PERFORMANCE_ CRITERION) OR
(model.param/MODEL.param > MAX_PRUNING_RATE):
    current_pruning_target = current_pruning_target +
PRUNING_STEP_SIZE
    # model pruning
    model = sparsity pruning(model,
    DATASET.train, GROWING_REG_EPOCHS)
    # remove the weights which have lowest Norm2 valu
    model = channel_prune_norm2(model, current_pruning_target, graph)
    model_tmp = model
    best_error = calc_error_mse(model, DATASET.val)

# retrain the pruned model with best validation error
    for e in range(RETRAIN_EPOCHS):
        model_tmp = train_one_epoch(model, DATASET.train)
        curr_error = calc_error(model_tmp, DATASET.val)
        if best_error < curr_error:
            model = model_tmp
            best_error = curr_error

# Output
model_output = model
The Reference Software of the Pruning Algorithm will be available at
the MPAI Git.
```

8.4 Reference Software

The Reference Software is released as Open Source Software with BSD 3-Clause Licence and can be downloaded from the MPAI Git.

9 EVC-UFV V1.0 Filter Examples

9.1 Introduction

This chapter provides the Neural Network weights obtained by applying the process specified in:

1. Design Procedure applied to
    1. The standard-definition to high-definition up-sampling, and
    2. High-definition to Ultra High-Definition up-sampling.
2. Complexity Reduction applied to the Neural Network of point 1, namely:
    1. The standard-definition to high-definition up-sampling, and
    2. High-definition to Ultra High-Definition up-sampling.

3. Procedure to generate an up-sampled image.

9.2 Test conditions

Table 1 provides the test conditions employed for the performance verification of the un-pruned and pruned up-sampling filters.

Table 1 – Test conditions for performance verification

Standard sequences	CatRobot, FoodMarket4, ParkRunning3.
Bits/sample	8 and 10 bit-depth per component.
Colour space	YCbCr with 4:2:0 sub sampling.
Encoding technologies	AVC, HEVC, and VVC.
Encoding settings	Random Access and Low Delay at QPs 22, 27, 32, 37, 42, 47.
Up-sampling	SD to HD and HD to UHD.
Metrics	BD-Rate, BD-PSNR and BD-VMAF
Deep-learning structure	Same for all QPs

Table 2 includes the performance results luminance only for Video sequences

1. unpruned and pruned up-sampling filters,
2. for SD to HD, HD to UHD and for SD to HD using the HD to UHD parameters
3. for videos that have been encoded with HEVC and VVC
4. in Low Delay (LD) and Random Access (RA) coding settings.

9.3 Performance results

Results show an impressive improvement for all coding technologies, and encoding options for all three objective metrics when compared with the currently used traditional bicubic interpolation.

Table 2 – Performance of the EVC-UFV Up-sampling Filter

		HEVC (LD)	VVC (LD)	HEVC (RA)	VVC (RA)
Unpruned	SD to HD (using own trained filter)	12.08%	13.74%	17.14%	22.5%
Unpruned	HD to UHD (using own trained filter)	4.05%	4.39%	6.29%	8.49%
Unpruned	SD to HD (using HD to UHD filter)	11.79%	13.45%	15.67%	20.38%
Pruned	SD to HD (using own trained filter)	12.2%	13.8%	17.3%	22.5%
Pruned	HD to UHD (using own trained filter)	6.0%	6.5%	6.0%	7.9%
Pruned	SD to HD (using HD to UHD filter)	11.6%	11.4%	15.3%	19.9%

Table 3 provides the same information for YUV sequences.

Table 3 – Performance of the EVC-UFV Up-sampling Filter

		HEVC (LD)	VVC (LD)	HEVC (RA)	VVC (RA)
Unpruned	SD to HD (using own trained filter)	U = 7.75 % V = 9.58%	U = 6.60% V = 7.93%	U = 10.90% V = 12.90%	U = 16.90% V = 1803%
Unpruned	HD to UHD (using own trained filter)	U = 7.83% V = 8.15%	U = 7.20% V = 7.41%	U = 9.39% V = 9.46%	U = 10.64% V = 10.92%
Unpruned	SD to HD (using HD to UHD filter)	U = 8.60% V = 10.51%	U = 6.18% V = 7.48%	U = 10.33% V = 12.17%	U = 14.52% V = 15.68%
Pruned	SD to HD (using own trained filter)	U = 8.38% V = 10.20	U = 6.47% V = 7.75%	U = 11.29% V = 13.20%	U = 16.75% V = 17.81%

Pruned	HD to UHD (using own trained filter)	U = 6.92% V = 7.24%	U = 6.81% V = 7.07%	U = 8.44% V = 8.63%	U = 9.99 % V = 10.39%
Pruned	SD to HD (using HD to UHD filter)	U = 7.84% V = 9.64%	U = 5.79% V = 7.07%	U = 9.52% V = 11.32%	U = 14.06% V = 15.16%

9.4 Testing the up-sampling procedure

The software at [MPAI Git]() enables users of this Technical Specification to:

1. Upload an SD or HD image
2. Select the type of weights – pruned or unpruned to be used to up-sample the provided image.
3. Download the up-sampled image.

Note that the image includes a small DeepCamera logo at the centre of the image

The number of parameters of the pruned filter is about 40% of the un-pruned filter.

The loss in performance of the pruned filter is less than 1% in BD-rate compared to the performance of the un-pruned filter.