MTJND: MULTI-TASK DEEP LEARNING FRAMEWORK FOR IMPROVED JND PREDICTION

Sanaz Nami[1,2], Farhad Pakdaman[2], Mahmoud Reza Hashemi[1], Shervin Shirmohammadi[3], Moncef Gabbouj[2]

[1]School of Electrical and Computer Engineering, University of Tehran, Iran
[2]Faculty of Information Technology and Communication Sciences, Tampere University, Finland
[3]School of Electrical Engineering and Computer Science, University of Ottawa, Canada

ABSTRACT

The limitation of the Human Visual System (HVS) in perceiving small distortions allows us to lower the bitrate required to achieve a certain visual quality. Predicting and applying the Just Noticeable Distortion (JND), which is a threshold for maximum unperceived level of distortions, is among the popular ways to do so. Recently, machine learning based methods have been able to reduce bitrate even further by improving JND prediction accuracy. However, accurate modeling of JND is very challenging, as it is highly content dependent. Furthermore, existing datasets provide little information to learn the best parameters. To remedy this issue, we propose a multi-task deep learning framework that jointly learns various complementary visual information. We design three separate methods and training strategies that jointly learn: (1) three JND levels, (2) visual attention map and a JND level, and (3) three JND levels and the visual attention map. We show that accumulating information from multiple tasks leads to a more robust prediction of JND. Experimental results confirm the superiority of our framework compared to the state-of-the-art.

Index Terms—Just Noticeable Distortion (JND), multi-task learning, visual attention, perceptual video coding

1. INTRODUCTION

Emerging media technologies such as Ultra High Definition (UHD) video streaming, Virtual Reality (VR), and Cloud Gaming (CG) have very high bandwidth and storage requirements. Therefore, it is desired to compress the video as much as possible, while maintaining the visual quality, by using Perceptual Video Coding (PVC) techniques [1][2][3]. These techniques eliminate perceptual redundancies by considering the characteristics of the Human Visual System (HVS) such as Visual Attention (VA), Regions of Interest (RoI), and our ability to observe small distortions.

One of the most efficient PVC approaches is to exploit Just Noticeable Distortion (JND) [1][4][5]. JND-based methods determine the maximum distortion level that still cannot be perceived by HVS, and use it as a threshold to remove the perceptual redundancies. This leads to the reduction of bitrate while maintaining visual quality.

JND-based methods have become popular recently, often with data-driven approaches. Datasets such as MCL-JCI [6], MCL-JCV [7] and VideoSet [8] contain various images/videos, and the Quality Factor (QF) or the Quantization Parameter (QP) associated with their JND levels. These datasets are collected via extensive subjective evaluations. Using MCL-JCI, Liu et al. [9] proposed a deep learning based binary classifier for images, which decides if a distorted image is visually different from its uncompressed reference. Tian et al. [4] also used MCL-JCI, which contains between 3 to 7 JND levels for each image, depending on the image content. Authors proposed a Convolutional Neural Network (CNN)-based model to predict the range of JND levels, and used Support Vector Regression (SVR) to estimate the number of JND levels. Finally, an algorithm is designed to find the QF for each JND level. Takeuchi et al. [10] also proposed an SVR-based JND model, based on Video Multimethod Assessment Fusion (VMAF), Structural Similarity (SSIM), and Peak Signal-to-Noise Ratio (PSNR). Zhang et al. [11] trained a CNN-based JND predictor on VideoSet, that estimates Video-wise JND by considering the spatial and temporal information. Nami et al. [12] proposed a JND-based Perceptual (JUNIPER) coding method, by jointly considering JND and saliency levels. This method was extended in [1] by proposing a JND mapping scheme that derives block-level JND levels from frame-level information. This enables a finer QP assignment in PVC. Wang et al. [13] segmented the image into different regions and established a region weighting model based on region color contrast. Then, a JND model is proposed by considering the weighting model and texture features. Shen et al. [14] trained a deep learning-based model to estimate the structure for each image. Then, they calculated the overall JND with a combination of three visual factors including luminance, contrast, and structure.

While JND-based methods perform well in image/video encoding, accurate prediction of JND levels is still a challenging research topic. The main reasons are that (1) JND models are trained based on rather small datasets, and (2) these datasets provide limited information (i.e., a single value representing the JND level), which is not sufficient to guide the network towards the best model parameters. Developing larger JND datasets involves hiring many subjects viewing tens or hundreds of video scenes, each with up to 51 qualities, and is very costly. However, one can leverage the useful information in relevant tasks to augment the information and improve the learning performance. Multi-Task Learning (MTL) [15] provides the means for joint learning of two or more tasks, which although may be different, have some similarities and can lead to a better learning. MTL has been used in similar image processing tasks, such as VA modeling [16][17] and image quality assessment [18][19], but has not yet been studied for JND prediction.

Motivated by the above-mentioned intuitions, the main goal of this research is to study joint learning of JND prediction with similar tasks, in a multi-task data-driven approach, to improve the accuracy of JND prediction. We consider two tasks, one with similar information, and one with complementary information. Using these two tasks, we design three MTL solutions as follows: (1) Joint learning of three JND levels (maximum number of levels provided by existing video datasets). While existing solutions train separate models to predict each JND level, these tasks are highly correlated and can help each other. (2) Joint learning of one JND level and the task of VA modeling. While VA is different from JND prediction and could be challenging to use in MTL, it provides complementary information that can push the training performance beyond those of one task. (3) Joint learning of the three JND levels and the VA

Accepted for presentation in IEEE ICIP 2023.

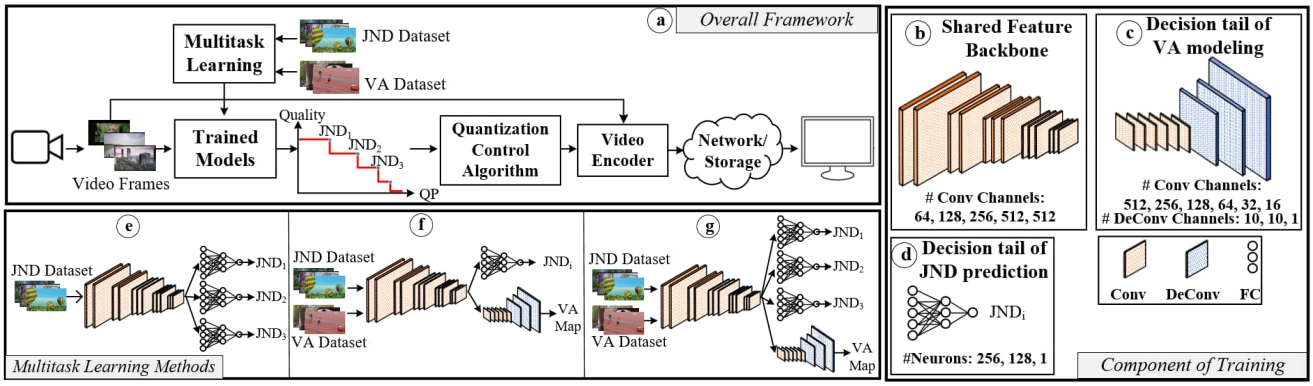

Fig. 1. The proposed framework and its components. (a) overall framework. (b) shared feature backbone. (c) decision tail for visual attention modeling. (d) decision tail for JND prediction. (e) MT_3LJND. (f) MT_1LJND_VA. (g) MT_3LJND_VA.

modeling. Our MTL framework exploits the correlations among the tasks and extracts effective features which improve the accuracy of each task. Project data will be available at https://github.com/sanaznami/MTJND. The contributions of the paper are as follows:

- We introduce an MTL-based framework for joint learning of JND prediction and VA modeling. We demonstrate that learning corelated tasks improves the generalization ability by learning a richer feature space. We develop three different MTL methods to jointly learn different tasks. To our best knowledge, this is the first study of MTL for JND prediction.
- Since there are no datasets that include labels for both tasks, we design training strategies to enable joint learning of tasks with different labels and different datasets, without the need for generating common datasets.
- Experimental results on two different JND datasets (for videos and images) demonstrate the effectiveness of our framework for both images and videos.

The remainder of the paper is organized as follows. The proposed framework is described in Section 2. The evaluation results are discussed in Section 3 and finally, Section 4 concludes the paper.

2. PROPOSED FRAMEWORK

In a single task learning approach, each task of JND prediction or VA modeling is developed independently of other tasks. Issues with such a paradigm are (1) various complementary visual perception modalities – although correlated – are not considered in the development of the model. (2) The existing JND datasets are rather small (too expensive to make larger), and training based on them can lead to overfitting, (3) JND datasets provide a single label for a rather complicated task, which is not sufficient for accurate modeling, especially given the limited number of samples. In an MTL, all tasks can be developed together. This means that more information can be used in the training process to regularize the training towards more effective features.

Fig. 1(a) depicts the overall framework, where a perception model based on JND and VA datasets is integrated into the video compression pipeline. The model is used to optimize the video encoder, via a quantization control algorithm. Fig. 1(e) to (g) show three different proposed MTL methods, which are as follows: (1) learning 3 JND levels together (MT_3LJND), (2) Learning one JND level together with the task of VA modeling (MT_1LJND_VA), and (3) Learning 3 JND levels together with the task of VA modeling (MT_3LJND_VA). Each of these models consist of a shared feature backbone to extract an efficient representation (Fig. 1(b)), and have a number of specific layers; i.e., decision tails (Fig. 1(c) and (d)). Details of the architecture and training process are discussed next.

2.1. Network Architecture

As illustrated in Fig. 1, our network consists of the feature backbone between tasks and the decision tails for each task. The feature backbone's architecture is motivated by the VGG network [20] and is altered experimentally to perform well in the MTL framework. The decision tails are designed according to each task's requirements, and also finalized experimentally.

The shared feature backbone is depicted in Fig. 1.(b), consisting of 13 Convolutional (Conv) layers with kernels of 3×3 pixels, which are activated by the Rectified Linear Unit (ReLU) function. The number of channels for the 13 layers is set to 64, 64, 128, 128, 256, 256, 256, 512, 512, 512, 512, 512, and 512, respectively. Zero padding in used to obtain an output of the same size as the input. A Max Pooling (MP) layer with kernels of 2×2 pixels, and a stride size of 2×2 is used after each 2 or 3 Conv layers.

JND prediction tail uses a decision tail depicted in Fig. 1.(d). There are 3 Fully Connected (FC) layers, activated by ReLU. These 3 FC layers include 256, 128, and 1 neurons, respectively.

Visual attention modeling tail uses the tail in Fig. 1.(c), consisting of 6 Conv layers with 512, 256, 128, 64, 32, and 16 channels followed by ReLU, and 3 DeConvolution (DeConv) layers with 10, 10, and 1 channels to construct the output the same size as the input.

2.2. Loss Function

Mean Squared Error (MSE) is commonly used for regression tasks such as JND prediction [11][21]. However, we observed that Mean Absolute Error (MAE) performs better in our MTL framework, perhaps as it is less affected by outliers [21]. Hence, MAE, L_J, is used as the loss function for JND prediction. Inspired by [16], cross entropy, L_{VA}, is selected as loss function for VA modeling.

Let $\{xj_t, jnd_t^i\}_{t=1}^{n}$ and $\{xa_t, va_t\}_{t=1}^{m}$ represent two different datasets of n and m training samples for the two tasks of JND prediction and VA modeling, respectively. Here, xj_t and xa_t are the t^{th} training images for JND prediction and VA modeling tasks, respectively. Also, jnd_t^i is the i^{th} JND level ground truth for the t^{th} image, and va_t represents the ground truth VA map for the t^{th} input. Furthermore, θ_S is the representation of all weights in the feature backbone and θ_J and θ_{VA} are the weights of the JND prediction and

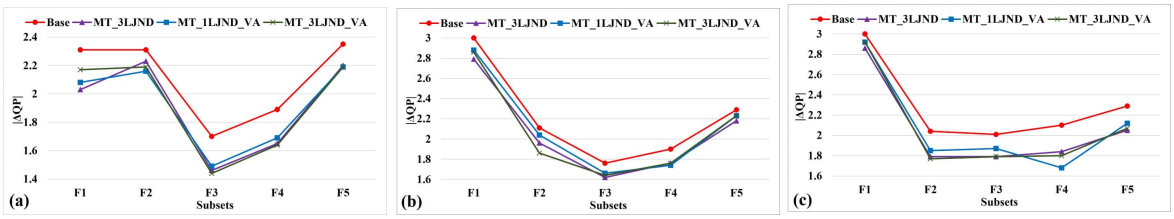

Fig 2. Testing loss of the 3 JND levels on VideoSet. (a) JND_1, (b) JND_2, (c) JND_3

the VA modeling tails, respectively. These weights are updated based on the two loss functions in (1) and (2), respectively, where $FJ(.)$ denotes the prediction function for JND prediction, and $FA(.)$ for VA modeling. Next subsections detail how these loss functions are used for the training of each of the proposed models.

$$L_J^i = \frac{1}{n}\sum_{t=1}^{n}\left|jnd_t^i - FJ\left(xj_t;\theta_S,\theta_J^i\right)\right|, \qquad i = 1,2,3 \tag{1}$$

$$L_{VA} = -\frac{1}{m}\sum_{t=1}^{m}\left[va_t \log FA(xa_t;\theta_S,\theta_{VA}) + (1-va_t)\log\left(1-FA(xa_t;\theta_S,\theta_{VA})\right)\right] \tag{2}$$

2.3. Multi-task Learning

MT_3LJND learns 3 JND levels together, on VideoSet dataset [8], which has 3 JND levels based on a QP value ranging from 1 to 51, for 220 5-second video sequences. According to Fig.1(e), MT_3LJND takes $\{xj_t\}_{t=1}^{n}$ as input for the training. Then, extracted features from the feature backbone are fed into the three identical decision tails of JND prediction. This method outputs 3 JND levels $\{\widehat{jnd}_t^i\}_{t=1}^{n}$. Here, a joint loss function is defined for MT_3LJND as shown in (3), where $\lambda_1, \lambda_2, and\ \lambda_3$ are three nonnegative weights.

$$L_{MT\_3LJND} = \lambda_1 L_J^1 + \lambda_2 L_J^2 + \lambda_3 L_J^3 \tag{3}$$

MT_1LJND_VA is trained to jointly learn predicting one JND level, and modeling the VA. Since there are no datasets available that contain both tasks' information, we design a training strategy that uses two different datasets for the two tasks and iterates the training over the two tasks. For JND prediction, the VideoSet dataset is used [8]. For the task of VA modeling, the large-scale eye-tracking database of videos (LEDOV) [22] is used which includes fixation points for 538 videos. According to Fig.1(f), MT_1LJND_VA takes $\{xj_t\}_{t=1}^{n}$ and $\{xa_t\}_{t=1}^{m}$ as training inputs. Then, extracted features from the feature backbone are fed into the decision tails of the JND prediction and VA tasks. Finally, this method outputs the i^{th} JND level and the VA map, referred to as $\{\widehat{jnd}_t^i\}_{t=1}^{n}$ and $\{\widehat{va}_t\}_{t=1}^{m}$, respectively. The training steps are:

i. The weights of the shared feature backbone are initialized with the first 13 Conv layers of the VGG network, which was pre-trained on the ImageNet dataset [23]. Weights of decision tails are initialized based on the Glorot uniform initialization [24].
ii. The visual attention model is trained over the LEDOV dataset, with the loss defined in (2). We name the parameters of the feature backbone and the visual attention tail after this stage as θ_S^1, and θ_{VA}^1, respectively.
iii. Next, the JND prediction task is trained over VideoSet, using the loss defined in (1) and starting with θ_S^1. After training, the new shared parameters, θ_S^2, and the parameters of the decision tail of JND, θ_J^1, are determined.
iv. Starting with the last achieved shared parameters, steps ii and iii are repeated until a desired solution is achieved.

In experiments, these steps are repeated for 2 iterations, 500 epochs for JND training and 100 epochs for VA training, at each iteration.

MT_3LJND_VA is a combination of the first two methods, where the prediction of 3 JND levels is learned jointly with the modeling of VA. VideoSet and LEDOV datasets are used for JND prediction and VA modeling, respectively. According to Fig.1(g), the MT_3LJND_VA method takes $\{xj_t\}_{t=1}^{n}$ and $\{xa_t\}_{t=1}^{m}$ as training inputs, and outputs 3 JND levels and the attention. The training procedure is very similar to MT_1LJND_VA, except that in step iii (3) is used instead of (1), to account for all three JND levels. The training parameters are updated based on Equations (3) and (2).

3. EXPERIMENTAL RESULTS

In this section, the performance of the proposed methods is evaluated. The proposed methods were implemented on Keras [25], and trained on a workstation with NVIDIA GeForce RTX 2080 Ti, with 11GB of graphics memory. As mentioned in 2.3, MT_1LJND_VA and MT_3LJND_VA models are trained for 2×(500+100) epochs. All other models are trained for 1000 epochs.

Performance of JND Prediction: The three proposed MTL methods are trained on VideoSet to predict JND levels. For fair evaluations, a similar single-task JND prediction method was trained as the ***baseline***, which consists of the same feature backbone and decision tails as MT_3LJND, but with one tail instead of three.

Similar to previous works [4][9], we randomly divide the dataset into five subsets (F_1 to F_5). To ensure that the proposed methods generalize well, the methods are trained with 5-fold cross validation. The results are presented in Figs. 2. (a), (b), and (c), where the *x*-axis represents the five subsets, and the *y*-axis represents |ΔQP|. It is observed that all three multi-task methods gain a |ΔQP| smaller than the baseline method. This suggests that information from the second task helped the main task of JND prediction, by learning a richer feature space. Also, it is observed that MT_3LJND and MT_3LJND_VA gain a better performance compared to MT_1LJND_VA. This may be because of more correlated tasks.

To simplify the evaluations and without lack of generalization, from this point forward we continue the results with the predictors trained on F_4, as the only predictor. We further analyze the performance of the methods using |ΔPSNR|, |ΔVMAF|, |ΔSSIM| and |ΔMSSSIM| in Table I. |ΔPSNR| is the absolute PSNR difference between the image encoded with the ground truth JND level, and the one encoded with the predicted JND (smaller |ΔPSNR| is better). The same is true for other metrics. From Table I, it is observed that the baseline method has the largest values and MT_3LJND_VA gains the smallest values for all metrics. Furthermore, Fig. 3 shows the training and validation loss over different epochs, for one JND (single task), and two and three JND levels (MTL). It can be observed that adding more tasks further reduces the training, and

Table I. Mean absolute prediction difference (smaller is better), on the VideoSet dataset

	|ΔPSNR|			|ΔVMAF|			|ΔSSIM|*10^{-3}			|ΔMSSSIM|*10^{-3}		
	JND_1	JND_2	JND_3	JND_1	JND_2	JND_3	JND_1	JND_2	JND_3	JND_1	JND_2	JND_3
Baseline (single task)	0.85	0.92	0.98	1.91	3.91	5.42	1.78	3.72	6.29	2.53	4.14	6.43
MT_3LJND	0.83	0.84	0.9	1.89	3.7	5.03	1.74	3.54	5.81	2.44	3.94	5.89
MT_1LJND_VA	0.84	0.83	0.87	1.94	3.71	4.83	1.74	3.55	5.59	2.42	3.93	5.64
MT_3LJND_VA	**0.8**	**0.78**	**0.86**	**1.84**	**3.4**	**4.76**	**1.66**	**3.16**	**5.37**	**2.3**	**3.47**	**5.41**

Table III. Compression Performance (G)

QF	75	50	25
PJND [4]	4.62	**5.35**	**5.23**
JUNIPER [12]	4.50	5.26	4.45
MT_3LJND	**4.68**	**5.35**	5.14

Table II. Comparing the proposed method (MT_3LJND) with two existing approaches, on the MCL-JCI dataset

	|ΔQF|				|ΔPSNR|				|ΔVMAF|				|ΔSSIM|*10^{-3}				|ΔLPIPS|			LOA	R^2
	JND_1	JND_2	JND_3	Avg	JND_1	JND_2	JND_3	Avg	JND_1	JND_2	JND_3	Avg	JND_1	JND_2	JND_3	Avg	JND_1	JND_2	JND_3		
PJND [4]	9.4	6.8	**6.0**	7.4	0.93	0.78	**0.99**	0.9	1.53	2.37	**4.83**	2.91	1.57	2.94	6.70	3.74	0.02	0.03	0.05	0.88	0.952
JUNIPER [12]	12.8	6.7	**6.0**	8.5	1.19	0.87	1.13	1.06	2.29	2.54	**4.83**	3.22	2.49	2.77	**5.87**	3.71	0.03	0.03	0.05	0.85	0.951
MT_3LJND	**6.4**	**5.5**	6.2	**6.03**	**0.55**	**0.62**	1.06	**0.74**	**1.10**	**1.91**	5.06	**2.69**	**1.08**	**2.37**	6.90	**3.45**	**0.01**	**0.02**	**0.05**	**0.90**	**0.971**

more importantly the validation loss.

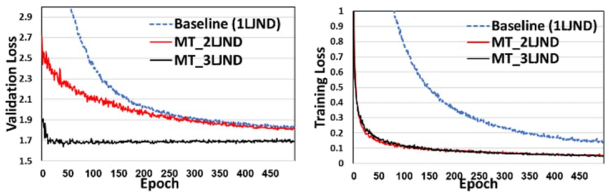

Fig 3. Comparing the learning performance with 1 to 3 JND levels

To better place the proposed MTL-based approach among existing solutions, the proposed methods are compared with [4] and [12] in the terms of |ΔQF|, |ΔPSNR|, |ΔVMAF| and |ΔSSIM| in Table II. Both of these methods [4] and [12] are evaluated on the MCL-JCI dataset, with JND levels based on QF ($QF \in [0, 100]$) values instead of QP ($QP \in [0, 51]$) which are used in the VideoSet dataset. To enable comparison, we fine-tune the MT_3LJND method, which was already trained on VideoSet, on images of MCL-JCI. To do so, we freeze the first 11 Conv layers, and train the remaining Conv layers and FC layers for another 500 epochs, starting from the weights trained on VideoSet.

As Table II shows, MT_3LJND achieves the best |ΔQF| for JND_1 and JND_2, and very close to the best for JND_3. The average |ΔQF| for MT_3LJND is 6.03, which is notably smaller than those of competing methods (7.4, and 8.5, respectively). Moreover, the more accurate JND modeling of the proposed method is reflected in terms of quality difference with the ground truth JND, in Table II. In all cases MT_3LJND achieves either the best performance or very close to the best. Delta Learned Perceptual Image Patch Similarity (|ΔLPIPS|) [26] in table is reported as a metric closely correlated to subjective quality. The close to zero |ΔLPIPS| reconfirms the visual similarity of the prediction. MT_3LJND also obtains the highest Level Overlapping Area (LOA) [4]. This is a measure of the overall prediction accuracy of all three JND levels. Finally, MT_3LJND achieves the highest R-Squared (R^2), computed for JND_1.

Compression Performance: As Fig. 1(a) shows, the trained models are deployed to optimize the compression pipeline, by steering the quantization parameter w.r.t the JND levels. To evaluate the compression performance, we use the same methodology as in [27]. The single metric Gain (G=ΔBitrate /ΔPSNR) defined in [27] jointly considers visual quality and bitrate saving. According to Table III, MT_3LJND obtains larger value of G in QF of 75 and 50 and very close to the best in QF of 25. This means our method can save more bitrates while maintaining visual quality as much as possible. It is worth noting that after training, the proposed MTL methods are used only for the prediction of the JND task and are not that different from competing tasks in terms of complexity or deployment cost.

Performance of Visual Attention: The main goal of this paper is the accurate prediction of JND levels, and we do not claim any contributions for VA modeling. However, Table IV summarizes the VA modeling performance, to provide insights on how MTL affects the VA modeling. The baseline is the same network as the one used in the MTL framework, only with a single tail for VA. It is observed that similar to the JND task, VA also benefits from the MTL framework; however, the achieved gain might be smaller compared to the JND task. This might be due to the fact that VA datasets already provide a more informative label (a map, instead of a single scalar as in JND datasets) to guide the single-task learning.

Table IV. Performance of VA modeling in single and multi-task

	Cross-Entropy	KLD	Correlation	MAE
Baseline	0.062	0.055	0.376	0.018
MT_3LJND_VA	**0.056**	**0.047**	0.397	0.019
MT_1LJND_VA	0.059	0.051	**0.398**	**0.017**

4. CONCLUSION

This paper presented a MTL framework for accurate JND prediction. Three different CNN-based methods have been proposed that learn to jointly predict a JND level, either with other JND levels, or with VA. It was demonstrated how joint learning of a corelated task helps to learn a better feature space, and further reduces the prediction error. It was observed that (1) both VA modeling and JND prediction (for a different JND level) can improve the performance as a second task, (2) adding to the number of tasks, e.g., co-learning of more JND levels together, improves the performance, and (3) more correlated tasks, such as joint learning of different JND levels, leads to more gain compared to a different task, such as VA modeling, and (4) VA modeling also improves in MTL framework, but the gain is limited compared to the JND prediction task. Extensive experimental results on two JND datasets (MCL-JCI and VideoSet) show that the proposed MTL framework outperforms existing methods on various evaluation metrics.

ACKNOWLEGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No [101022466], and from the NSF-Business Finland Center for Visual and Decision Informatics (CVDI) Advanced Machine Learning for Industrial Applications (AMaLIA) under Grant 97/31/2023.

5. REFERENCES

[1] S. Nami, F. Pakdaman, M. R. Hashemi, and S. Shirmohammadi, "BL-JUNIPER: A CNN-Assisted Framework for Perceptual Video Coding Leveraging Block-Level JND," *IEEE Trans Multimedia*, pp. 1–16, 2022.

[2] H. Ahmadi, S. Zadtootaghaj, F. Pakdaman, M. R. Hashemi, and S. Shirmohammadi, "A Skill-Based Visual Attention Model for Cloud Gaming," *IEEE Access*, vol. 9, pp. 12332–12347, 2021.

[3] S. H. Bae, J. Kim, and M. Kim, "HEVC-Based Perceptually Adaptive Video Coding Using a DCT-Based Local Distortion Detection Probability Model," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3343–3357, 2016.

[4] T. Tian, H. Wang, L. Zuo, C. C. J. Kuo, and S. Kwong, "Just noticeable difference level prediction for perceptual image compression," *IEEE Transactions on Broadcasting*, vol. 66, no. 3, pp. 690–700, 2020.

[5] X. Zhang, H. Wang, and T. Tian, "Perceptual Video Coding with Block-Level Staircase Just Noticeable Distortion," in *IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 4140–4144.

[6] L. Jin *et al.*, "Statistical study on perceived JPEG image quality via MCL- JCI dataset construction and analysis," in *International Symposium on Electronic Imaging*, 2016, pp. 1–9.

[7] H. Wang *et al.*, "MCL-JCV: A JND-based H.264/AVC video quality assessment dataset," in *IEEE international conference on image processing (ICIP)*, 2016, pp. 1509–1513.

[8] H. Wang *et al.*, "VideoSet: A large-scale compressed video quality dataset based on JND measurement," *J Vis Commun Image Represent*, vol. 46, pp. 292–302, 2017.

[9] H. Liu *et al.*, "Deep Learning-Based Picture-Wise Just Noticeable Distortion Prediction Model for Image Compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 641–656, 2020.

[10] M. Takeuchi *et al.*, "Perceptual Quality Driven Adaptive Video Coding Using JND Estimation," in *Picture Coding Symposium (PCS)*, 2018, pp. 179–183.

[11] Y. Zhang, H. Liu, Y. Yang, X. Fan, S. Kwong, and C. C. J. Kuo, "Deep Learning Based Just Noticeable Difference and Perceptual Quality Prediction Models for Compressed Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1197–1212, 2022.

[12] S. Nami, F. Pakdaman, and M. R. Hashemi, "Juniper: A jnd-based perceptual video coding framework to jointly utilize saliency and JND," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2020, pp. 1–6.

[13] C. Wang, Y. Wang, and J. Lian, "A Superpixel-Wise Just Noticeable Distortion Model," *IEEE Access*, vol. 8, pp. 204816–204824, 2020.

[14] X. Shen, Z. Ni, W. Yang, X. Zhang, S. Wang, and S. Kwong, "Just Noticeable Distortion Profile Inference: A Patch-Level Structural Visibility Learning Approach," *IEEE Trans Image Process*, vol. 30, pp. 26–38, 2021.

[15] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," *IEEE Trans Knowl Data Eng*, vol. 34, no. 12, pp. 5586–5609, 2021.

[16] X. Li *et al.*, "DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3919–3930, 2016.

[17] Q. Zhang, S. Wang, X. Wang, Z. Sun, S. Kwong, and J. Jiang, "A Multi-Task Collaborative Network for Light Field Salient Object Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1849–1861, 2021.

[18] L. Xu *et al.*, "Multi-Task Rank Learning for Image Quality Assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 1833–1843, 2017.

[19] F. Li, Y. Zhang, and P. C. Cosman, "MMMNet: An End-to-End Multi-Task Deep Convolution Neural Network with Multi-Scale and Multi-Hierarchy Fusion for Blind Image Quality Assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4798–4811, 2021.

[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations, ICLR*, International Conference on Learning Representations, ICLR, 2015.

[21] S. Bosse, D. Maniry, K. R. Müller, T. Wiegand, and W. Samek, "Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2018.

[22] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "DeepVS: A Deep Learning Based Video Saliency Prediction Approach," in *In Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 602–617.

[23] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int J Comput Vis*, vol. 115, pp. 211–252, 2015.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[25] F. Chollet and Others, "keras-team/keras: Deep Learning for humans," 2015. https://github.com/keras-team/keras

[26] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[27] Q. Jiang, Z. Liu, S. Wang, F. Shao, and W. Lin, "Toward Top-Down Just Noticeable Difference Estimation of Natural Images," *IEEE Transactions on Image Processing*, vol. 31, pp. 3697–3712, 2022.