Wireless Image Retrieval at the Edge

Mikolaj Jankowski, Deniz Gündüz and Krystian Mikolajczyk
Imperial College London

***Abstract*—We study the image retrieval problem at the wireless edge, where an edge device captures an image, which is then used to retrieve similar images from an edge server. These can be images of the same person or a vehicle taken from other cameras at different times and locations. Our goal is to maximize the accuracy of the retrieval task under power and bandwidth constraints over the wireless link. Due to the stringent delay constraint of the underlying application, sending the whole image at a sufficient quality is not possible. We propose two alternative schemes based on digital and analog communications, respectively. In the digital approach, we first propose a deep neural network (DNN) aided retrieval-oriented image compression scheme, whose output bit sequence is transmitted over the channel using conventional channel codes. In the analog joint source and channel coding (JSCC) approach, the feature vectors are directly mapped into channel symbols. We evaluate both schemes on image based re-identification (re-ID) tasks under different channel conditions, including both static and fading channels. We show that the JSCC scheme significantly increases the end-to-end accuracy, speeds up the encoding process, and provides graceful degradation with channel conditions. The proposed architecture is evaluated through extensive simulations on different datasets and channel conditions, as well as through ablation studies.**

***Index Terms*—Deep learning, Internet of Things, image retrieval, joint source-channel coding, person re-identification.**

I. Introduction

INTERNET of Things (IoT) devices are becoming increasingly widespread. These small specialized computers are present in offices, streets, and homes. Their main goal is to continuously sense their environment, and send the measurements through a wireless channel to an edge server, which performs data collection and further processing. Typical approach in most IoT applications is to convey all the measurements from the IoT devices to an edge server, where state-of-the-art machine learning algorithms are used to analyse the collected data. However, in some applications, the volume of the measurement data (e.g., images, videos or LIDAR data) is large, and transmitting it to the server at the required quality may not be feasible within the limited latency requirements, e.g., in autonomous driving, surveillance, drones, etc. On the other hand, as the computational capabilities of IoT devices advance, they can process the data locally before offloading it to a server. In some cases the desired inference tasks can be carried out locally, which is beneficial as the IoT devices have access to the original data, rather than its quantized version at the edge server, due to the lossy compression and transmission over the wireless channels.

This paper was presented in part at the 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) [1].

This work was supported in part by the European Research Council (ERC) through Starting Grant BEACON under Grant 677854 and in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/N007743/1, Grant EP/S032398/1, and Grant EP/T023600/1.

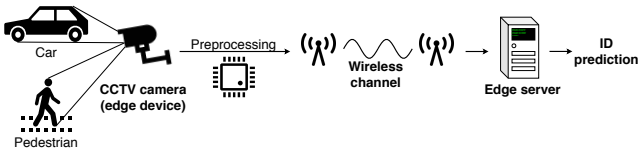

Fig. 1: An illustration of the retrieval problem at the edge. A CCTV camera takes a picture of a pedestrian or a car, and processes the image locally to obtain a low-dimensional signature, which is then sent through a wireless channel to an edge server that performs identification based on a large database it has access to.

In this work, we study machine learning at the wireless edge. In particular, we focus on distributed inference over a wireless channel, where a centrally-trained algorithm is deployed on IoT devices to perform inference over-the-air. One of the machine learning tasks for which remote inference is essential is retrieval. In autonomous vehicles, drones, or in surveillance systems, agents try to identify objects, vehicles, or humans in their environment through their sensory data. The goal in image retrieval is to identify a query image of a person or a vehicle recorded locally by matching with images stored in a large database (gallery), typically available at the edge server (cf. Fig. 1). We emphasize that the retrieval task cannot be performed locally at the edge device regardless of its computational power. This is because the centralized database is available only at the edge server, hence, some sensory data has to be transmitted to the edge server. The fundamental question we want to answer in this paper is what part or function of data must be transmitted, and how.

A trivial response to these questions would be to convey the image to the edge server at the best quality possible. The server first reconstructs the image, and performs the retrieval task with a state-of-the-art retrieval algorithm. Note, however, that a significant part of the image content may not be relevant for the retrieval task, therefore the original image is not needed at the server. Indeed, novel approaches to retrieval employ deep neural networks (DNNs) as feature encoders that map input images to a low-dimensional feature space, such that vectors extracted from the same identities are similar, despite different views or occlusions. Accordingly, we employ DNNs for extracting features that are then transmitted over the wireless link.

We propose two approaches to convey the feature vectors to the edge server. In the conventional "digital" approach, feature vectors are first compressed, and encoded with a channel code for reliable transmission. The features that are most relevant for the retrieval task are extracted and transmitted depending

on the capacity and the reliability of the channel between the edge device and the server. To improve the efficiency of this approach, we design a retrieval-oriented image compression scheme, which compresses the feature vectors depending on the available bit budget. This "separate" data compression and channel transmission scheme assumes reliable communication over the channel. Such scenario is typically difficult to achieve in practice, especially for short blocklengths considered in this work, imposed by the strict delay limitations. Alternatively, we consider a joint source and channel coding (JSCC) approach, where the feature vectors are directly mapped into channel input symbols, and the noisy channel output is used by the server to retrieve the most relevant images, without involving any explicit channel code. This can be considered as "analog" communication since the feature vectors are not converted into bits at any stage. For the JSCC approach, we employ an architecture based on DNNs, similar to the novel DeepJSCC [2], [3], which has recently been introduced for wireless image transmission. Our results show that the JSCC scheme can outperform the highly optimized feature compression scheme even if we assume the availability of capacity-achieving channel codes for the digital scheme. To the best of our knowledge, this is the first work to study image retrieval over a wireless channel. Our specific technical contributions can be summarized as follows:

- We propose a novel retrieval-oriented image compression scheme, which combines a retrieval baseline with a feature encoder, followed by scalar quantization and entropy coding. To estimate the distribution to be used for the entropy coder, we introduce a density model based on a Gaussian mixture.
- We propose an autoencoder-based architecture and training strategy for robust JSCC of feature vectors, generated by a retrieval baseline, under noisy, fading, and bandwidth-limited channel conditions.
- We perform extensive evaluations under different signal-to-noise ratio (SNR) and bandwidth constraints, and show that the JSCC scheme outperforms the digital approach even with capacity-achieving channel codes. Moreover, its performance exhibits graceful degradation when the test and training SNRs do not match. The JSCC scheme is shown to outperform its digital counterpart also over fading channels, even if we assume the availability of channel state information for the digital scheme only.
- We evaluate the proposed schemes on various surveillance tasks, and show that the performance close to the noiseless bound can be achieved even under very harsh SNR and bandwidth constraints, whereas the digital approach falls short of this performance even with idealistic capacity-achieving channel codes.
- Our results show that, in general, it is not possible to separate inference tasks from the communication scheme, and the end-to-end performance can be improved significantly by designing the communication and learning algorithms jointly. We provide a comprehensive analysis of different architectures and training strategies that will serve as solid baselines for future research in wireless edge learning.

In this paper we extend our previous work [1] by considering different wireless channel models to show the generalization of our method and we validate the methods by extensive evaluations on new datasets. We provide a comparison of different architectures and training methods for wireless image retrieval. In our digital model we introduce a new, simpler, but equally effective density model based on a Gaussian mixture.

II. Related work

A. Machine Learning at the Wireless Edge

With the increasing computational capabilities of edge devices, many recent studies consider executing machine learning tasks across edge devices. Many of these works focus on the training stage, which is particularly challenging due to the distributed nature of data available at edge nodes, and the typically limited communication resources (please see [4]–[9] and references therein).

Instead, in this work, we focus on the inference phase, assuming that the training can be run centrally. This approach requires centralized availability of the training data. Prior works on distributed inference at the wireless edge have focused on classification tasks using DNNs. Authors of [10]–[15] suggest splitting neural network architectures into two parts to reduce the computational workload at the edge device. In this work we do not consider computational limitations of the device, and perform the forward pass over the DNN locally, at the edge device, which was shown in [15] to reduce the bandwidth necessary to transmit the information for the classification task. Digital schemes for distributed inference, e.g. [11], [13], limit the amount of information (e.g., the number of bits) that can be conveyed to the edge server, but ignore the energy and latency cost of communications, and potential errors that may be introduced. However, in practice, reliable transmission of the feature vectors, even if they are highly compressed, requires an accurate estimate of the channel state at the edge device, and a very reliable error correction code. However, not only such a separate approach is suboptimal, but also channel codes introduce significant error probability at short blocklengths, especially in the absence of accurate channel state information. Analog schemes based on JSCC have recently been considered in [12], [14], [15], and they were shown to outperform separate approaches, but they focus on the classification task using low-resolution images. This significantly reduces the amount of information to be transmitted, as the task is to distinguish between a finite set of known classes. In contrast, in the retrieval and re-identification tasks, we require high resolution images, and have to cope with unknown set of identities, thus the feature vectors have to convey significantly more information. Unlike in the classification, the retrieval task cannot be performed locally at the edge device due to its limited computational resources and data transmission to the edge server is needed.

B. Person and Vehicle Retrieval

Person and vehicle retrieval tasks have been extensively studied [16]–[22]. They share the same motivation to allow for a better and more reliable recognition of people and

vehicles, mainly targeting surveillance applications. The most successful recent approaches for image retrieval problems are based on convolutional neural networks, and recent techniques include part classifiers [16], [17], creating bias-invariant feature vectors [18], [22], using attention models [20], and analyzing images at different scales [23], [24]. Despite the popularity of triplet loss in both areas [19], [25], designs based on softmax cross-entropy have also been successfully implemented [21].

C. Joint Source-Channel Coding (JSCC)

According to Shannon's separation theorem [26], performing source and channel coding separately achieves theoretical optimality guarantees in the asymptotic infinite blocklength regime. This theorem holds under average power constraint and a single-letter additive distortion constraint, e.g., average mean-square error between the samples of the input and output sequences. However, in practice, we are limited by finite blocklengths due to complexity and latency constraints; and JSCC is known to outperform separate schemes in practical scenarios. Many JSCC schemes have been proposed [27]–[29], but these have not found application in practice as they are too complex and specific to the underlying source and channel distributions. Moreover, they do not provide sufficient improvement to justify the introduced increase in the system complexity, as well as the loss of modularity. More recently, JSCC schemes based on autoencoders [30], which are DNNs aimed at unsupervised data coding, have been introduced [2], [3], [31], [32], and are shown to provide comparable or better performance than state-of-the-art digital schemes.

JSCC for remote inference problems is much less studied. Distributed hypothesis testing problem over a noisy communication channel has been recently introduced in [33] using an information theoretic formulation and considering the type II error exponent as the performance measure. Here, the goal is to make a decision on the joint distribution of the samples observed by a remote observer and those observed by the decision maker. Similarly to our setting, the observer communicates to the decision maker over a noisy channel. It is shown that, while the optimality of separation holds for the problem of testing against independence, where the alternative hypothesis is the product of the marginal distributions of the remote and local samples, separation is suboptimal in general, when testing against arbitrary joint distributions.

III. Methods

In this work we propose two approaches for performing retrieval over wireless channels: digital (separate) and JSCC (joint) approaches. In both cases, we consider the transmission of the feature vectors, which are a low-dimensional representation of identities of the items to be retrieved e.g., humans, vehicles (Section III-B), and have to be sent over bandwidth-limited wireless channels. Due to the channel limitations, features cannot be transmitted in a lossless fashion, and have to be compressed. The recovered noisy feature vectors at the receiver are compared to the feature vectors of images previously collected from other edge cameras, called the *gallery*, in order to find the nearest neighbour.

A. Channel Model

We assume that the edge device is connected to the edge server through an additive white Gaussian noise (AWGN) channel. We consider static as well as slow fading channel. For both approaches presented in this work, we assume that the channel model is known during training, and remains the same during inference.

The AWGN channel is characterized as follows: given a channel input vector $\mathbf{x} \in \mathbb{C}^{\quad}$, consisting of complex channel input symbols G_i, the output $\mathbf{y} \in \mathbb{C}^{\quad}$ is given by $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $I_i \quad \mathcal{CN}(0, f^2)$ are the independent and identically distributed (i.i.d.) elements of the noise vector $\mathbf{z} \in \mathbb{C}^{\ -i} = 1, \dots, \ $. An average power constraint is imposed on the input vectors, such that $\frac{1}{\ } \sum_{i=1}^{\ } |G_i|^2 \quad P = 1$; which, in the case of a static AWGN channel, translates into a maximum received SNR of $\mathrm{SNR} = 10 \log_{10}\left(\frac{1}{f^2}\right)$ in dB scale.

In the slow fading scenario, we consider a single-tap Rayleigh fading channel model, where all the transmitted symbols experience the same channel gain. That is, given the channel input vector $\mathbf{x} \in \mathbb{C}^{\ }$, the corresponding output vector $\mathbf{y} \in \mathbb{C}^{\ }$ is given by $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\quad \mathcal{CN}(0, f^2)$ and $I_i \quad \mathcal{CN}(0, f^2)$ are drawn from independent zero-mean complex normal distributions with variances f^2 and f^2, respectively. We impose the same average input power constraint of $P = 1$ as in the AWGN case. For each transmitted feature vector we use a single gain , which characterizes the *slow fading* behaviour. The maximum average SNR is evaluated by $\mathrm{SNR} = 10 \log_{10}\left(\frac{f^2}{f^2}\right)$ dB, while for all the experiments shown in this paper we set $f^2 = 1$, which corresponds to the same average received power as in the static AWGN channel model.

B. Retrieval Baseline

Following the state-of-the-art retrieval methods [17], [21] we employ the ResNet-50 network [34], pretrained on ImageNet [35], for feature extraction. This ensures that similar results can be expected in different setups. In more detail, we use ResNet-50 with batch normalization (BN) layers applied after each convolutional layer. As input, we use images resized to a common $256 \quad 128$ resolution with bicubic interpolation for person datasets and $128 \quad 128$ resolution for vehicle datasets. For the last layer we use average pooling across all the feature maps, which results in a 2048-dimensional feature vector. During training we use stochastic gradient descent (SGD) with a learning rate of 0·01 and a momentum of 0·9. We also apply l_2 regularization, weighted by $5 \quad 10^{\ 4}$ to the ResNet-50 parameters. We refer to this architecture as the *feature encoder*.

C. Digital Transmission of Compressed Feature Vectors

This approach is based on the assumption that a certain number of bits can be reliably conveyed to the edge server for each image. In practice, however, this is highly challenging to achieve. Ultra-reliable channel codes require large blocklengths even in the static AWGN setting, and accurate channel estimation and feedback in the slow-fading case. In our simulations, we assume capacity-achieving channel codes, which will serve as a bound on the performance of practical digital schemes.

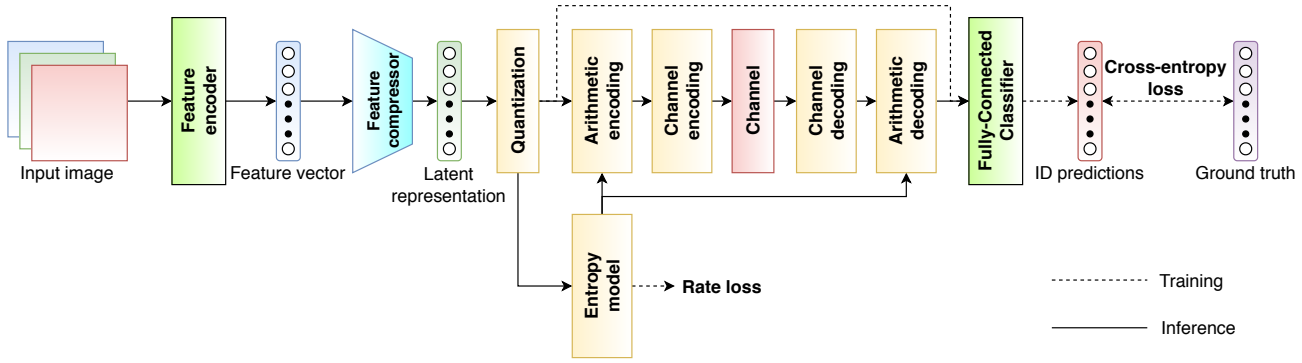

Fig. 2: The digital transmission scheme. Input is transformed into a feature vector, which is compressed using a DNN. At the receiver, latent representation is classified into IDs to compute the loss during training only. Arithmetic coding and channel coding is bypassed during training.

An overview of the proposed digital scheme is shown in Fig. 2. We first extract features using the retrieval baseline described in Section III-B as feature encoder. The resulting feature vector is compressed into as few bits as possible through lossy compression followed by arithmetic coding. The compressed bits are then channel coded, with introduced structured redundancy to combat channel impairments.

The lossy feature compressor consists of a single fully-connected layer for dimensionality reduction, followed by quantization. On the receiver side we use the quantized latent representation as a feature vector, which is passed through a fully-connected layer for ID classification. Note that the IDs and their classification are used for calculating the loss during training only. During retrieval, the IDs are not known and the feature vectors are used for nearest neighbour search. This has been shown to perform well in the re-ID community [16]–[22].

To enable an end-to-end differentiable approach, we utilize the well-known quantization noise [36] to model the quantization process. Specifically, instead of rounding the latent representation to the nearest integer, in the training phase we add the uniform noise to each element of the latent representation as follows:

$$\&^{1}\mathbf{z}^{o} = \mathbf{z} + \mathsf{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \tag{1}$$

where $\&^{1\,o}$ is the approximated quantization operation, $\mathbf{z}$ is the latent representation, and $\mathsf{U}^{1}(-\frac{1}{2}, \frac{1}{2})^{o}$ is the uniform noise vector. This formulation ensures a good approximation of quantization during training, whereas we perform rounding to the nearest integer during inference.

In order to optimize the arithmetic coder, we estimate the distribution of the quantized outputs. We assume that the elements $@$ of vector $\mathbf{q} = \&^{1}\mathbf{z}^{o}$ are i.i.d. with some probability mass function (PMF) $?^{1}@^{o}$. To model this PMF, we propose a simple yet flexible solution using a mixture of Gaussians. We first approximate $?^{1}@^{o}$ as a continuous-valued probability density function $?_{2}{}^{1}@^{o}$ as follows:

$$?_{2}{}^{1}@^{o} = \sum_{:=1}^{\tilde{O}} U_{:} \frac{1}{f_{:}\sqrt{2c}} 4^{\frac{1}{2}\left(\frac{@-`_{:}}{f_{:}}\right)^{2}} \tag{2}$$

where $\tilde{O}$ is the number of mixtures, $f_{:}$ are mixture scales, $`_{:}$ are mean values, and $U_{:}$ are the corresponding mixture weights. In our experiments we set $\tilde{O} = 9$, which we empirically found to perform the best. Then, in order to evaluate our PMF $?^{1}@^{o}$ at discrete values $@ \in \mathsf{Z}$, we integrate $?_{2}{}^{1}@^{o}$ over $@ \pm \frac{1}{2}$ to obtain:

$$?^{1}@^{o} = \int_{@-\frac{1}{2}}^{@+\frac{1}{2}} ?_{2}{}^{1}G^{o}\mathrm{d}G = {}_{2}\left(@ + \frac{1}{2}\right) - {}_{2}\left(@ - \frac{1}{2}\right) \tag{3}$$

where ${}_{2}$ is the cumulative density function of the distribution $?_{2}{}^{1}@^{o}$.

We remark that, here we learn the distribution of the quantized feature vectors, but unlike recent works [37], [38] , we do not consider adaptive probability model and do not introduce another neural network to predict parameters $\{U_{:}, `_{:}, f_{:}\}$ of the mixture. The reason for that is the proposed simple model performs sufficiently well, and we want to avoid introducing any communication overhead by sending additional parameters per image. Instead, we use the available bandwidth for sending quantized feature vectors only.

With the model presented above, we can easily estimate the PMF of the quantized vector $\mathbf{q}$, which can be directly used to feed the arithmetic coding engine in the test phase, but also to evaluate the average approximate entropy over the dataset in our loss function, which we define as a weighted sum of two objectives:

$$! = ;_{24} + \log_{2} ?^{1}\mathbf{q}^{o} \tag{4}$$

where $;_{24}$ is the cross-entropy between the predicted class (identity) and the ground truth for the retrieval task. The second component of the loss function corresponds to the empirical Shannon entropy of the quantized vector, representing the average length of the output of the arithmetic encoder. Such formulation allows for a smooth transition between the retrieval accuracy and number of bits necessary to send the feature vector in a lossy fashion. Moreover, minimizing the entropy term is equivalent to maximizing the likelihood of $?^{1}@^{o}$, which corresponds to increasing the certainty of our model, and allows a satisfactory fit of our approximated

distribution to the true underlying distribution of the discrete symbols.

We apply the same settings discussed in Section III-B to train the feature encoder, the fully-connected classifier and the density model. We train the whole network for 20 epochs, reduce the learning rate to 0•001 and train for further 30 epochs. We initialize our mixture parameters as follows: $U_: = \frac{1}{}$, $`_: = 0$, $f_: = :^2$- $: = 1$-2-••• . To ensure the convergence during training, in the first epochs we prioritize the $;_{24}$ loss term by setting the weight parameter $\__8$ at epoch 8 as:

$$\__8 = \min\left(\_\frac{8}{20}\text{-}\_ \text{ - } 8 = 1\text{-}\bullet\bullet\bullet\text{-} \text{ -}\right) \quad (5)$$

where i 20 is the total number of epochs. In our experiments we use $\_$ 2 »10 ^{5}-10 ^{2¼}, and = 50.

In the inference phase we use the arithmetic encoding engine to transmit the information with a channel code. Note that any channel code can introduce errors, there is therefore an inherent trade-off between the compression rate and the channel coding rate under a given constraint on the channel bandwidth, i.e., the number of channel symbols that can be transmitted to the edge server per image pixel. Compressing the feature vector further leads to increased distortion, and hence, reduced retrieval accuracy, but also allows to introduce more redundancy, and hence, increased reliability against noise. In general, the optimal compression and channel coding rates depend on the retrieval accuracy-compression rate function of the compression scheme and the error-rate of the channel code. To simplify this task, we assume capacity-achieving channel codes, which provides an upper bound on the performance that can be achieved by any digital scheme that uses the above architecture.

D. JSCC of Feature Vectors

In this section, we propose an alternative JSCC approach, called JSCC AE, and illustrated in Fig. 3. We use the baseline feature encoder as before to produce the feature vector for a given query image. The feature vector is mapped directly to the channel input symbols via a multi-layer fully-connected JSCC encoder (Fig. 4a). We set the dimensionality of the channel input vector to 2 real symbols, which corresponds to the available channel bandwidth of complex values. In this work we consider small values of modeling stringent delay constraints of the underlying surveillance applications. This low-dimensional representation is normalized to satisfy the average power constraint of % = 1, and transmitted over the AWGN channel. The noisy channel output vector at the receiver is mapped back to the high-dimensional feature space by a JSCC decoder (Fig. 4b). The distance between the query feature vector and the feature vectors stored in the gallery set is calculated to find the nearest neighbours.

In order to train our network, the most straightforward strategy would be to perform end-to-end training, taking images from the dataset as an input, and training both the feature encoder and the JSCC autoencoder jointly, in an end-to-end fashion, with cross-entropy loss between the ID predictions and the ground truth (as shown in Fig. 3). However, our experimentation in Section IV-F shows that this approach leads to suboptimal performance. Alternatively, we propose traininig each component of the network separately at first, and, once the feature encoder and the JSCC autoencoder are pretrained individually, they are combined and trained jointly. Therefore, our training strategy, which we refer to as $)_{1\text{-}2\text{-}3}$, consists of three steps: feature encoder pretraining ($)_1$), JSCC autoencoder pretraining ($)_2$), and end-to-end training ($)_3$). In the first step, $)_1$, we attach a single fully-connected layer at the end of the feature encoder that maps 2048-dimensional feature vectors directly to the ID predictions. We pretrain the feature encoder for 30 epochs with a batch size of 16, using cross-entropy between the ID predictions and the ground truth as the loss function. In the second step, $)_2$, we freeze the pretrained feature encoder, and use it to extract features from all the images in the training dataset. We use these features as inputs to the proposed autoencoder network. We train the autoencoder using the $!_1$-loss between the feature vectors and the vectors reconstructed by the JSCC decoder. It is trained with SGD for 200 epochs with a learning rate 0•1, reduced to 0•01 after 150 epochs, and momentum of 0•9. We apply $!_2$ regularizer to the autoencoder model, weighted by 5 10 ^{4}. Finally, in the third step, $)_3$, we train the whole network jointly, the autoencoder and the feature encoder, for 30 epochs, using the cross-entropy loss with a learning rate 0•001, and for further 10 epochs with a learning rate of 0•0001, applying the same optimizer and $!_2$ regularization as in the previous two steps.

Along with $)_{1\text{-}2\text{-}3}$ we evaluate four alternative training strategies. The first one, denoted by $)_3$, corresponds to the end-to-end training of the entire network (feature encoder + JSCC autoencoder + classifier) in a single training step. The second method, $)_{1\text{-}2}$, consists of the feature encoder pretraining, $)_1$, followed by the JSCC autoencoder training, $)_2$ to reconstruct feature vectors with $!_1$ as the distortion measure. This method corresponds to using a JSCC scheme whose goal is to reconstruct the feature vector as reliably as possible without taking into account the accuracy of the retrieval task. After $)_2$, the feature encoder and the autoencoder are combined as in Fig. 2, but the joint training step, $)_3$, is not performed. The third method, $)_{1\text{-}3}$, consists of the feature encoder pretraining, $)_1$, followed by joint training of the entire network, $)_3$. Finally, $)_{1\text{-}3}$ ¸ $!_1$ approach is different from the $)_{1\text{-}3}$ in that it combines the cross-entropy loss and $!_1$ loss, in the joint training phase.

Note that, we opted for an architecture that employs a distinct feature encoder and a separate JSCC autoencoder to transmit the feature vector over the channel. We have then trained these components in multiple training steps. It is possible to introduce a simpler architecture with a single JSCC encoder at the edge device that maps the query image to the channel input vector. Thus, no decoding is required at the receiver, and the retrieval task is directly performed using the noisy channel symbols. To compare our method to this straightforward approach, we introduce JSCC FC, which follows the same structure as in Fig. 3, except that the JSCC encoder is replaced by a single fully-connected layer and the JSCC decoder is removed. We train the whole network end-

Fig. 3: The architecture and training of JSCC of feature vectors for wireless image retrieval. The feature vector is directly mapped to channel inputs. Received noisy signal is decoded and processed by a fully-connected layer to obtain ID predictions, which are then compared to the ground truth by the cross-entropy loss.

(a) JSCC encoder (b) JSCC decoder

Fig. 4: Proposed JSCC encoder and JSCC decoder architecture for the JSCC scheme illustrated in Fig. 3. At the encoder, dimensionality reduction is performed by the first fully-connected layer, which is inverted at the decoder.

to-end for 50 epochs with cross-entropy loss, learning rate of 0.01, reduced to 0.001 after 30 epochs, and a momentum of 0.9. We also apply l_2 regularization, weighted by $5 \cdot 10^{-4}$, to all the parameters, including ResNet-50, feature encoder and fully-connected classifier.

IV. Results

In this section we evaluate the performance of the proposed JSCC AE and JSCC FC architectures, and compare with that of the digital scheme presented in Section III-C, as well as the ideal channel scenario with unlimited channel resources, where full, noiseless feature vectors can be transmitted over the channel. We first discuss the experimental setup and the dataset used for the evaluations.

A. Experimental Setup

For the JSCC AE and JSCC FC schemes we vary channel SNRs for training, between $SNR_{CA08=}$ = −12dB and $SNR_{CA08=}$ = 1 dB, which corresponds to zero noise power. Training and test SNRs are the same unless stated otherwise. In the digital scheme, we experiment with different dimensionality of the latent representation, between 64 and 512, estimate and minimize its entropy in the training phase by varying the value of parameter λ. In the testing phase we perform rounding to the nearest integer on each element of the latent representation and arithmetic coding, which is based on the probabilistic model learned by the entropy estimator, as described in Section III-C. This model assigns a probability estimate to each quantized symbol, which is then passed to the arithmetic encoder. We note that the proposed digital scheme is a variable-length encoder. Therefore, for a given fixed communication rate to the server, one has to determine the λ coefficient that meets the rate constraint for each image. Instead, we fix the λ coefficient and calculate the average number of bits required to encode the latent representations of the test images. We then evaluate the corresponding channel SNR to deliver these many bits to the receiver, assuming capacity-achieving channel codes. This is the upper bound on the real performance as practical codes are far from the capacity bound in the short blocklength regime. This model may correspond to sending multiple images together, and hence, the performance is determined by the average rate across many test images, rather than their individual rates.

For digital transmission over a fading channel, we consider two scenarios. In the first one, we assume perfect channel state information available at both the transmitter and the receiver. Then, for each query image and a corresponding random channel gain, we identify the λ parameter that results in a bit rate that is as close as possible from below to the corresponding channel capacity. Then, we find the average accuracy across many random queries and channel conditions, following the underlying fading distribution. In the second scenario, we fix the λ parameter, and for each query image and the corresponding random channel condition, we compare the required bit rate of the query image and the channel capacity. If the capacity is lower than the bit rate required by the compression scheme, we assume the transmission is failed. We then calculate the fraction of successful transmissions and multiply it by the average accuracy of the queries whose compressed feature vector can be successfully transmitted, for a given λ. Note that, there is a trade-off between the accuracy loss due to compression and the outage over the channel. The higher λ values results in more compact representations of the feature vector, and hence less accurate retrieval performance even if they can be successfully conveyed to the server. Higher λ values relaxes the compression constraint, but may result in higher loss over the channel. Note that, we report only the results for the λ values that lead to the highest average accuracy for each average SNR.

To train our model we used NVIDIA GeForce RTX 2080Ti GPU. A single end-to-end training of our digital model took approximately 35 minutes, which was similar to the training time of the JSCC FC. For JSCC AE, the training took approximately 20 minutes, 3 minutes, and 30 minutes for λ_1, λ_2, and λ_3, respectively. Please note that λ_1 has to be performed only once, as this step does not depend on the channel model.

(a) Person re-ID CUHK03 - AWGN

(b) Person re-ID CUHK03 - fading

(c) Person re-ID Market-1501 - AWGN

(d) Person re-ID Market-1501 - fading

(e) Car re-ID VeRi - AWGN

(f) Car re-ID VeRi - fading

Fig. 5: Performance comparison of the proposed three schemes over AWGN and slow fading channels for a range of channel SNRs and bandwidth = 64. Our JSCC AE scheme achieves the best retrieval accuracy over the whole range of tested SNRs and for all three re-ID image retrieval datasets.

B. Datasets

In order to measure the performance of the retrieval task, we employ three widely used datasets:

CUHK03 [39] is a benchmark for person retrieval that contains 14096 images of 1467 identities taken from two different camera views. The dataset was captured with six surveillance cameras and each identity within the dataset is represented by an average of 4.8 images per each of the two camera views. We use the labeled variant of the dataset, where each image of the pedestrian was manually cropped by a human.

Market-1501 [40] contains 32217 images of 1501 pedestrians taken from a total of six cameras in front of a supermarket at Tsinghua University. Five out of six cameras are high-resolution cameras and the remaining one is low-resolution. Training and testing splits proposed by the authors contains 12936 and 19732 images, respectively. 750 identities are additionally selected as a query set containing 3368 images (maximum of 6 per person). The dataset is different from CUHK03 in that it contains junk images capturing only partial pose and distractors presenting small fragments of pedestrian appearance or irrelevant objects.

VeRi [41], [42] is a vehicle retrieval dataset. It contains over 50000 images of 776 vehicles captured by 20 cameras within 24 hours over the area of $1 km^2$. Each identity is captured by 2-18 cameras in different viewpoints, occlusions, resolutions, and lighting conditions. All the images within the dataset are annotated with attributes, brands and colors, but in this work we do not utilize this information, and focus on retrieving the identity only based on the image.

The evaluation measure for all the datasets is the top-1 retrieval accuracy, which calculates the fraction of correct IDs at the top of the ranked list retrieved for each query.

C. Performance for Different Methods

We plot the accuracy achieved by various schemes as a function of the test SNR in Fig. 5. For these experiments we use the bandwidth of 64, which corresponds to the transmission of 64 complex symbols through the channel. One can see that JSCC AE outperforms the digital scheme in all considered scenarios. For CUHK03 dataset the digital approach is not able to recover the noiseless accuracy even at SNR = 15dB, whereas the proposed JSCC AE scheme obtains accuracy close to the ideal channel baseline at around 10dB for the AWGN channel. JSCC FC follows JSCC AE very closely, but the increase in the performance provided by the autoencoder is visible for all the SNRs considered, which proves the superiority of the proposed architecture in comparison to the relatively simple JSCC FC. The lower accuracy of JSCC FC may stem from the fact that the noise directly affects the low-dimensional feature vector, while the autoencoder-based scheme introduces certain level of denoising, which improves the feature estimates at the receiver. In Fig. 5a, we also show that feature decoding is not beneficial for the digital scenario. An alternative scheme, which we called Digital w/ decoding (capacity achieving), follows the same training strategy as discussed in Section III-C, but we further introduce a fully-connected decoder. This decoder is placed before the fully-connected classifier, and maps low-dimensional quantized latents back to the original, 2048-dimensional feature vector space. We show that this decoding step brings no improvement to the digital scheme performance, compared to the scenario without decoding. This result was consistent across all the datasets, but to avoid clutter we show it only in Fig. 5a. Another observation is that the relative performances of the three schemes are similar for all the datasets considered, while JSCC FC seems to perform worse for the Car VeRi dataset, and even surpassed by the digital scheme at SNR = 10dB.

Fading channels introduce additional perturbation to the channel symbols, reducing the accuracy of all the proposed approaches. Similarly to the AWGN channel, JSCC AE achieves the best performance across all three datasets and the average SNR values considered in this paper. The digital scheme performs worse when the channel state information is not available (which is also the case for the JSCC schemes). We have also included the performance of the digital scheme when perfect channel state information is available. We observe that even in this case the proposed JSCC AE scheme outperforms the digital alternative. JSCC FC closely follows JSCC AE at the low SNR regime, but its performance saturates to a level significantly below that of JSCC AE, and even below the digital scheme for the CUHK03 dataset. This result further validates the denoising interpretation of the autoencoder structure in JSCC AE, which becomes even more critical in recovering the noisy feature vector in the presence of channel fading. Fading not only applies random attenuation to the received signal strength, but also random rotations in the complex plain, which makes it very difficult for the receiver to recover the features for correct retrieval without any channel state information. We note that, while the digital scheme suffers significant performance loss in the absence of the channel state information, JSCC AE seems to perform reasonably well. We can argue that the autoencoder learns to mitigate the effect of random fading despite the lack of explicit pilot signals. We also provide the performance of JSCC AE, when perfect channel state information is available at the receiver. The JSCC decoder first divides the received signal by the channel gain: $\frac{y}{z} = x + \frac{n}{z}$, and reconstructs the input feature from this. In this scenario, our method is able to recover the noiseless bound at a reasonable SNR of 15dB, and the gap between JSCC AE w/ CSI and the digital approach grows even further.

D. Performance for Different Bandwidths

In this experiment we investigate the effect of the channel bandwidth on the retrieval performance for the person retrieval CUHK03 dataset, achieved by the JSCC AE scheme. We emphasize that the previously considered bandwidth of $k = 64$ is extremely limited, corresponding to extremely low-latency communications, which may be essential for many surveillance and security applications. The top-1 accuracy as a function of the channel SNR is plotted in Fig. 6 for different channel bandwidth values of $k = 64, 128, 256$, and 512. It shows that the accuracy and robustness increases significantly with the bandwidth, but the relative gain becomes smaller as we approach the original feature vector dimension.

(a) AWGN channel

(b) Fading channel

Fig. 6: Accuracy as a function of the channel SNR for different channel bandwidths. Higher bandwidth introduces more robustness against the channel noise.

For the fading channel, it is visible in Fig. 6b that the proposed JSCC AE scheme without the channel state information is not able to recover the original accuracy even for a significant bandwidth, and reaches a plateau at around SNR = 12dB. As pointed out in Section IV-C, this may stem from the fact that our approach cannot fully cancel the effect of the variable channel gain. Channel estimation and feedback techniques can be utilized to mitigate the impact of random channel fading, as shown in Section IV-C.

E. Graceful Degradation

In this section we evaluate the behaviour of our models on the CUHK03 dataset when the training and test SNRs do not match. In the experiments with the digital scheme, we assume that capacity-achieving channel codes are in use, and the quality of the channel is always estimated correctly. However, in practice, digital approaches suffer from the *cliff effect*, which results in a sharp decrease in the performance when the channel condition is worse than the channel state for which the channel code is designed. If the code rate is above the current channel capacity, it is known that true error probability converges to 1 [43].

On the other hand, unlike digital models, analog transmission schemes are known to achieve graceful degradation when we are interested in the end-to-end reconstruction quality [2]; that is the average reconstruction quality smoothly decreases as the channel conditions become worse. This behaviour is quite beneficial, since we do not have to train multiple autoencoders, one for each channel SNR value, or even introduce channel estimation and feedback feature if the performance does not critically depend on applying the same training and testing SNRs. In the previous sections we showed the best possible accuracy for a specific SNR, which means each data point corresponds to a model trained specifically for that target SNR. In Fig. 7 we show that graceful degradation can be achieved with the proposed JSCC AE architecture, and it is not necessary to train a separate model for every SNR value. Instead, we can take a model trained with a moderate $\mathrm{SNR}_{\mathrm{CA08}}=$ and apply it to a wide range of $\mathrm{SNR}_{\mathrm{CA08}}$ in the $\mathrm{SNR}_{\mathrm{C4BD}}$ inference time at the expense of a moderate loss in accuracy. To the best of our knowledge, this is the first time graceful degradation is demonstrated for the inference as opposed to the average reconstruction quality that is typically considered in the literature.

Note that the approach trained without noise ($\mathrm{SNR}_{\mathrm{CA08}}=$ 1 dB) is not robust against the channel noise. Therefore, its accuracy decreases much faster than for the networks trained under different noise levels, yet it still shows graceful degradation as the channel noise increases.

F. Training Strategy

In this section we show the superiority of $)_{1\text{–}2\text{–}3}$ training strategy by comparing to the alternative training methods introduced in Section III-D. Note that, for the fairness of the comparison, we perform the first step of the training, which is the feature encoder pretraining, only once for $)_{1\text{–}2\text{–}3}$, $)_{1\text{–}2}$ $)_{1\text{–}3}$, and $)_{1\text{–}3}$, ! $_{1}$.

The evolution of the cross-entropy loss over training epochs of the final joint training phase for different training strategies is shown in Fig. 8. In the experiment we used the bandwidth = 64 and SNR=0dB. The proposed three-step training allows to achieve much better final performance, as shown in Table I. Here, we also shown the top-5 recognition accuracy, i.e., the correct match was listed within the top 5 ranklist elements, and the mean average precision (mAP), which are standard evaluation measures for the retrieval tasks. Adding each training step increases the performance gradually, and there is a significant difference between $)_{1\text{–}3}$, and $)_{1\text{–}3}$, as well as between $)_{1\text{–}3}$ and $)_{1\text{–}2\text{–}3}$. Our three-step strategy outperforms all three alternatives by a large margin as it converges faster and achieves the smallest loss after the last epoch. As expected, it performs the worse, since it has to learn both the retrieval

(a) AWGN channel (b) Fading channel

Fig. 7: Accuracy achieved by the proposed JSCC AE scheme as a function of SNR_{CHBC} for different SNR_{CHAN} values for $M = 64$. JSCC AE achieves graceful degradation with the channel SNR as opposed to the digital scheme, which suffers from the cliff effect. Models trained at moderate SNR_{CHAN} values achieve relatively good performance for a wide range of test SNRs values.

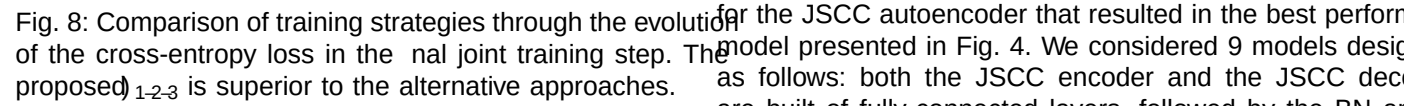

Fig. 8: Comparison of training strategies through the evolution of the cross-entropy loss in the final joint training step. The proposed $\theta_{1\text{-}2\text{-}3}$ is superior to the alternative approaches.

and robustness against the noise in a single training step with randomly initialized weights. Interestingly, the convergence of the $\theta_{1\text{-}3}, \theta_1$ seems to slightly outperform the convergence of the $\theta_{1\text{-}3}$, thanks to the additional loss term, which forces the reconstructed features to be similar to the original ones. However, while this seems to speed-up the convergence of the autoencoder network marginally, it does not affect the final performance. The reasonable performance of $\theta_{1\text{-}2}$ shows that θ_2 allows the autoencoder to produce good reconstructions of the feature vectors under noisy environment, but the gap between $\theta_{1\text{-}2}$ and $\theta_{1\text{-}2\text{-}3}$ indicate the necessity of joint training phase, θ_3, which maximizes the task performance. One may argue that our $\theta_{1\text{-}2\text{-}3}$ strategy is slower compared to the alternatives, nevertheless adding the autoencoder pretraining phase is negligible in comparison to the joint training phase (3min vs. 1hr).

TABLE I: Comparison of the retrieval performance for different training strategies.

Method	Top-1 accuracy	Top-5 accuracy	mAP
θ_3	0.225	0.409	0.195
$\theta_{1\text{-}3}$	0.312	0.533	0.286
$\theta_{1\text{-}3}, \theta_1$	0.317	0.536	0.287
$\theta_{1\text{-}2}$	0.330	0.557	0.306
$\theta_{1\text{-}2\text{-}3}$	0.392	0.602	0.351

G. Comparison of Different Models

In this section we present the results of architecture search for the JSCC autoencoder that resulted in the best performing model presented in Fig. 4. We considered 9 models designed as follows: both the JSCC encoder and the JSCC decoder are built of fully-connected layers, followed by the BN and activation layers. The only exceptions are the last layers in the JSCC encoder and the JSCC decoder which are without BN and activations. The first layer of the JSCC encoder maps 2048-dimensional features to $2k$ real-valued symbols, which eventually forms a k complex symbols transmitted over the channel. Similarly, the last layer of the JSCC decoder maps $2k$-dimensional vectors back to the original 2048-dimensional feature space and the remaining FC layers keep the dimension at $2k$. The evaluated architectures and results are shown in Table II. We select the models by starting from the baseline (denoted as A) from [1] and then removing or adding layers from the JSCC encoder and the JSCC decoder networks to explore the impact of depth on the overall performance. For each model we trained the network according to the three-step strategy described in Section III-D and performed

TABLE II: Person retrieval accuracy for the CUHK03 dataset achieved by different models at SNR = 0dB and = 64.

Model	# JSCC encoder layers	# JSCC decoder layers	Activation	MSE	Top-1 accuracy	Top-5 accuracy	mAP
A	3	3	Leaky ReLU	0.204	0.382	0.**602**	0.354
B	3	2	Leaky ReLU	0.222	0.391	0.597	0.354
C	3	4	Leaky ReLU	0.199	0.390	0.601	0.358
D	2	3	Leaky ReLU	0.202	**0.392**	**0.602**	**0.359**
D	4	3	PReLU	**0.181**	0.383	0.589	0.343
E	4	3	Leaky ReLU	0.208	0.383	0.598	0.356
F	1	1	N/A	0.207	0.387	0.592	0.352
G	2	2	Leaky ReLU	0.206	0.387	0.592	0.352
H	1	2	Leaky ReLU	0.207	0.386	0.593	0.353
I	2	1	Leaky ReLU	0.206	0.389	0.600	0.356

evaluation on the CUHK03 dataset at SNR = 0dB, = 64. We also show the mean squared error between the original feature vectors and their noisy reconstructions, after JSCC autoencoder pretraining $)_2$. The results show that the differences between the models are marginal. Model D, which corresponds to the architecture presented in Fig. 4 and was used in the rest of the paper, performs slightly better than the others in terms of final retrieval performance. This model was selected also due to its low computational cost, as it consists of only 5 fully-connected layers in total. We also used PReLU as the activation for the model variant D, and observed that even though it achieves better MSE in step $)_2$, it fails to provide a good generalization capabilities in the final step, as it overfits to the data. Please note that the model F, does not have the activation function, as the only layers in both the encoder, and the decoder are the last layers, therefore, as described above, the activation and BN are removed.

V. Conclusions

In this work, we have introduced the image retrieval problem over wireless channels in the context of the edge network, where wireless edge devices send queries of images over a bandwidth and power limited channel to an edge server that stores the image database, also called the gallery. We first introduced a digital approach, which is based on a novel retrieval-oriented deep image compression scheme, and applied it to feature vectors obtained from the feature encoder. Next, we proposed a JSCC-based scheme, where feature vectors are directly mapped to the channel symbols and decoded at the receiver. We showed the latter approach not only achieves a superior retrieval accuracy at a target channel SNR, but also provides graceful degradation with the test SNR when it does not match the training SNR. We further introduced JSCC FC, which is a simplified version of the proposed model and showed that decoding is necessary at the receiver to mitigate the effects of channel impairments. We also proposed a novel strategy for training our JSCC scheme, that can be adapted to other machine learning applications performed over noisy channels. Our strategy achieves superior performance for training the JSCC scheme. We have also performed an extensive ablation study of different architectures and training strategies and compared the alternatives under various performance measures for a wide range of different channel conditions. The results show the superiority of the proposed architecture and the joint training approach.

References

[1] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Deep joint source-channel coding for wireless image retrieval," in *IEEE Int'l Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, 2020, pp. 5070–5074.

[2] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. on Cognitive Comms. and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[3] D. B. Kurka and D. Gündüz, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[4] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. on Communications*, vol. 68, no. 1, pp. 317–333, Jan 2020.

[5] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. Schaar, "Machine learning in the air," *IEEE Journal on Selected Areas in Comms.*, vol. 37, no. 10, pp. 2184–2199, 2019.

[6] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.

[7] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *IEEE Int'l Symposium on Information Theory (ISIT)*, 2019, pp. 1432–1436.

[8] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. on Wireless Communications*, vol. 19, no. 5, pp. 3546–3557, 2020.

[9] M. Abad, E. Ozfatura, D. Gündüz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, May 2020.

[10] W. Shi, Y. Hou, S. Zhou, Z. Niu, Y. Zhang, and L. Geng, "Improving device-edge cooperative inference of deep learning via 2-step pruning," in *IEEE INFOCOM 2019 - IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 1–6.

[11] A. Eshratifar, A. Esmaili, and M. Pedram, "Bottlenet: A deep learning architecture for intelligent mobile cloud computing services," in *IEEE/ACM Int'l Symp. on Low Power Elec. and Design*, 2019, pp. 1–6.

[12] J. Shao and J. Zhang, "Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems," in *2020 IEEE Int'l Conf. on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[13] H. Li, C. Hu, J. Jiang, Z. Wang, Y. Wen, and W. Zhu, "Jalad: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution," in *IEEE Int'l Conf. on Parallel and Distributed Systems (ICPADS)*. IEEE, 2018, pp. 671–678.

[14] C. Lee, J. Lin, P. Chen, and Y. Chang, "Deep learning-constructed joint transmission-recognition for internet of things," *IEEE Access*, vol. 7, pp. 76547–76561, 2019.

[15] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Joint Device-Edge inference over wireless links with pruning," in *IEEE Int'l Workshop on Signal Proc. Adv. in Wireless Comms. (SPAWC)*, May 2020.

[16] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *CoRR*, vol. abs/1610.02984, 2016.

[17] B. He, J. Li, Y. Zhao, and Y. Tian, "Part-regularized near-duplicate vehicle re-identification," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 3997–4005.

[18] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose-invariant embedding for deep person re-identification," *IEEE Trans. on Image Processing*, vol. 28, no. 9, pp. 4500–4509, 2019.

[19] R. Kuma, E. Weill, F. Aghdasi, and P. Sriram, "Vehicle re-identification: an efficient baseline using triplet embedding," in *Int'l Joint Conf. on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–9.