POLITECNICO DI TORINO

Department of Electronics and Telecommunications

Master's degree in communications engineering

Simulation analysis of semantic communications applied to a human language use case

Supervisors

Prof. Carla Fabiana Chiasserini

Dr. Corrado Puligheddu

Eng. Roberto Fantini

Eng. Elisa Zimaglia

Candidate

Antonio Pio Grieco

2024

Thesis Project for the Master of Science in Communications Engineering

Study of Semantic Communications and Simulation of a Use Case

Antonio Pio Grieco

Prof. Carla Fabiana Chiasserini
Dr. Corrado Puligheddu
Eng. Roberto Fantini
Eng. Elisa Zimaglia

Department of Electronics and Telecommunications
POLITECNICO DI TORINO
Turin, Italy 2024

Abstract

Since their inception, the objective of digital communication systems has been to effectively and reliably transmit a series of bits representing a certain type of information.

Consequently, all engineering choices made in past years to construct such systems pursued this goal, overlooking the semantic significance of the information intended for transmission.

Semantic communications, viewed as a potential breakthrough in the Shannon paradigm, aim to convey the meaning of a message rather than accurately transmitting every symbol.

Recent advancements in machine learning have enabled the extraction of such information from signals, a development that has the potential to revolutionize signal processing techniques in the future.

This could significantly enhance the performance of telecommunication systems, whose design must be conceived with the aforementioned objectives in mind.

The purpose of this thesis is to provide an overview of semantic communications, analysing the current state of the art and the associated challenges.

In addition to the theoretical analysis, this thesis features an experimental component that simulates a textual semantic transmission in a realistic context using a 5G New Radio-based transmission chain. These experiments aim to make a comparison and to highlight the benefits of the new semantic paradigm over conventional communication systems.

Table of Contents

List of figures

List Of Tables

ACRONYMS

3GPP	Third Generation Partnership Project
ADNet	Attention-guided Denoising convolutional neural Network
AEP	Asymptotic Equipartition Property
AI	Artificial Intelligence
AR	Augmented Reality
ASR	Automatic Speech Recognition
BER	Block Error Rate
BLEU	Bilingual Evaluation Understudy
BLSTM	Bi-directional Long Short-Term Memory
BS	Base Station
CDL	Clustered Delay Line
CE	Cross-Entropy
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CSI	Channel State Information
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
ETSI	European Telecommunication Standards Institute
FDSD	Frechet Deep Speech Distance
FPN	Feature Pyramid Network
GAN	Generative Adversarial Network
GoP	Group of Pictures
GRU	Gated Recurrent Unit
H-ARQ	Hybrid Automatic Repeat Request
HMM	Hidden Markov Models
idd	Independent and Identically Distributed
IoT	Internet of Things
ISI	Inter-Symbol Interference
JSCC	Joint Source-Channel Coding
KDSD	Kernel Deep Speech Distance
KL	Kullback-Leibler
LDPC	Low-Density Parity-Check

LOS	Line-Of-Sight
LSTM	Long-Short-Term Memory
MDP	Markov Decision Process
MFCC	Mel-frequency Cepstral Coefficients
MI	Mutual Information
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MMSE	Minimum Mean Squared Error
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NLOS	Non-Line-Of-Sight
NLP	Natural Language Processing
NR	New Radio
PESQ	Perceptual Evaluation of Speech Quality
POLQA	Perceptual Objective Listening Quality Assessment
PSNR	Peak-Signal-to-Noise Ratio
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RCNN	Recurrent Convolutional Neural Network
RePN	Relation-Proposal-Network
RNN	Recurrent Neural Network
RPN	Region Proposal Network
RS	Reed-Solomon
SDR	Signal-to-Distortion Ratio
SER	Symbol-Error-Rate
SFC	Space Frequency Coding
SISO	Single-Input Single-Output
SM	Spatial Multiplexing
SSF	Scaled Space Flow
SSIM	Structural Similarity Index
SSW	Scale-Space Warping
STE	Straight-Through Estimator
STOI	Short-Time Objective Intelligibility
SVD	Single Values Decomposition
SVM	State Vector Machine
TBS	Transport Blocks
TDL	Tapped Delay Line
UE	User Equipment
VGG	Very deep convolutional network
VR	Virtual Reality
ZF	Zero Forcing

1. Introduction

This chapter provides an overview of semantic communications, from principles to challenges to be addressed.

It is worth mentioning that the concept of semantic communications was described over 70 years ago by W. Weaver, who in his seminal paper defined three levels of problems in communications: the technical level, concerning the accurate transmission of symbols; the semantic level, addressing how precisely said symbols convey semantic meaning; and the effectiveness level, concerning the effects resulting from such information exchange.

In contrast to the Shannon paradigm, which entails the transmission of every bit of information produced by the source, the key idea of semantic communications is to extract relevant features for a specific task at the receiver.

Consider for example, the case where the task to be performed is image recognition; the information source would not transmit bits representing the entire image but would instead extract the relevant features to represent the subject of the image, such as a human being or a specific object.

It is, therefore, straightforward to understand how omitting the background from the transmission would minimize the amount of data transmitted, thereby enhancing the system performance in terms of its wireless resource utilization or energy consumption, all without compromising the outcome of the task.

1.1 Information Theory Overview

Before exploring and defining semantic information, it is useful to briefly recall some concepts of classical information theory. However, given its vast scope, this review will focus only on those concepts closely related to semantic information theory.

In 1949, C. Shannon introduced the concept of information entropy [1], which employs a probabilistic approach to measure the amount of information in terms of bits.

Definition 1: Given a source X, represented by a discrete random variable that takes values from (x_1) to (x_n) with probabilities $p(x_1), p(x_2), \ldots, p(x_n)$, the source entropy $H(X)$, defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\, log\, p\,(x_i) \tag{1}$$

quantifies how many bits of information the source X produces on average.

Definition 2: Given a communication channel with input X, and output $\underline{Y}$, the mutual information between input and output can be expressed as:

$$I(X;Y) = H(Y) - H(Y|X) \tag{2}$$

Where:

- $H(Y)$ is the entropy of the output (Y)
- $H(Y|X)$ is the conditional entropy of (Y) given (X)

These concepts are of fundamental importance for formulating the Channel Capacity Theorem, also known as the Shannon-Hartley Theorem.

Theorem 1: The maximum achievable rate $C(X)$ of reliable transmission over a noisy channel, defined as:

$$C(X) = \max_{p(x)} I\,(X;Y) \tag{3}$$

is equal to the maximum of the mutual information between input and output of the channel, where $p(x)$ represents the probability distribution of the input symbols.

Furthermore, Shannon defined the asymptotic equipartition property AEP.

Property 1: For a sequence of i.i.d. random variables generated by a discrete memoryless source, the empirical distribution of the sequence converges to the true distribution of the source as the sequence length tends to infinity.

$$\lim_{n\to\infty} \Pr\left(\left|-\frac{1}{n}\log_2 P\,(X_1, X_2, \dots, X_n) - H(X)\right| > \epsilon\right) = 0 \quad (4)$$

In simpler terms, the AEP implies that as the sequence length grows, the average entropy converges to the entropy of the source with high probability.

Theorem 2: If (x_i) for $(i = 1, \dots, n)$ satisfies the AEP and $(H(X) < C)$, there exists a source-channel code with a probability of error $(p(\hat{x}_i \neq x_i) \to 0)$. Conversely, the error probability will be positive if the constraint on entropy is not met.

These concepts and theorems lay the groundwork for understanding and optimizing communication systems in various scenarios and applications.

Theorem 1 provides an upper limit for distortion-less transmissions.

Given a specified distortion (D^*), the minimum transmission information rate (R) can be described by the Lossy Source Coding Theorem, also known as the Rate-Distortion Theorem.

Theorem 3: For a given maximum average distortion (D^*), the rate-distortion function $R(D^*)$ represents the lower bound of the bitrate in transmission.

$$R(D^*) = \min_{D \le D^*} I\,(X;Y) \quad (5)$$

where $D = \sum p(x)p(u|x)d(x,y)$ with $d(x,y)$ being the distortion metric equal to 0 if $x = y$, as

$$R(D^*) = \min_{D \le D^*} I\,(X;Y) \le \min_{D=0} I\,(X;Y) = H(X) \quad (6)$$

Modern communication systems are based on Shannon's Separation Theorem, which distinguishes two stages:

1. Efficient data compression (source coding)
2. Mapping the source coded sequence into a channel coded sequence (channel coding)

1.2 Semantic Information Theory

1.2.1 Semantic Entropy

Entropy, as defined by Shannon, measures the information content based on the uncertainty of the source; however, it overlooks how to measure the amount of semantic information for a specific transmission task, where the transmission task is the task to be performed at the receiver after the information is received.

From the point of view of a traditional communication system, the semantic information extracted from the source information can be seen as a lossy compression; however, in the context of semantic communications, it represents a lossless compression, since the extracted information can fully serve the transmission task without performance degradation. Therefore, the transmission task makes possible to measure the importance of information, recalling the previous image classification example, the objects represented in an image are considered essential information while the background has a limited relevance to the specific transmission task.

In the past decades, researchers have tried to find a way to quantify semantic entropy; actually, this research area is still very active since no definitive definition has been provided. In the following the main contributions are summarized.

Definition 3: Carnap and Bar-Hillel [2] defined the semantic entropy by the degree of confirmation, which is:

$$H(H,\mathrm{e}) = -\log c(H,e) \tag{7}$$

where $c(H,e)$ is the degree of confirmation of the hypothesis H over the evidence e, where the hypothesis could be a message, and the evidence could be the knowledge.

Definition 4: Bao et al. [3] focus on the semantic entropy of a message or sentence s, defining the entropy as:

$$H(s) = -log_2\big(m(s)\big) \tag{8}$$

Where $m(s)$ is the logical probability of s, which is:

$$\mathrm{m(s)} = \frac{\mathrm{p(W_s)}}{\mathrm{p(W)}} = \frac{\sum_{\mathrm{w}\in\mathrm{W},\mathrm{w}|=\mathrm{s}} p(w)}{\sum_{w\in W} p(w)} \tag{9}$$

Where $W$is the symbol set of a classical source, $\vDash$ is the proposition satisfaction relation and $w \in (\mathrm{w} \in W, w| = s)$ is the set in which s is true, in the end $p(w)$ is the probability of w, which is equal to 1 if there is no background available.

Based on the Axiomatic Fuzzy Set theory, Liu et al. [4] defined the semantic entropy as:

$$H_{C_{j(\varsigma)}} = -\mathrm{D_j}(\varsigma)\log_2 \mathrm{D_j}(\varsigma) \tag{10}$$

where ς is a semantic concept, which can be treated as the transmission task, $\mu_\varsigma(\mathrm{x})$ is the membership degree for each $x \in X$, C_j is the j-th class and $\mathrm{D_j}(\varsigma) = \frac{\sum_{x\in X_{C_j}} \mu_\varsigma(\mathrm{x})}{\sum_{x\in X} \mu_\varsigma(\mathrm{x})}$ is the matching degree, which characterizes the entropy of the element x on the concept ς. Note that the overall semantic entropy can be obtained by summing up that of all the classes.

1.2.2 Semantic Channel

Classical telecommunication systems, relying on Shannon theorem, measure the distortion errors introduced by the channel by means of bit-error-rate BER or symbol-error-rate SER (engineering level error).

However, since semantic communications focus on the semantic content of a message instead of the bits composing it, a different approach to measure the distortion introduced by the communication channel is required.

In order to clarify the difference between the two types of errors, consider a transmission where the input message, e.g. *"vending machine"*, and the recovered output is different, e.g. *"vending machin"*: classical communication systems will detect an error, however from a semantic point of view, the input and output message can still convey the same meaning, ensuring that the subsequent task is performed correctly. Therefore, it does not make sense to evaluate the semantic transmission performance based on the BER or SER metrics in the context of a semantic

transmission: note that in this case an error would be such only if the meaning of the message is changed (e.g. “blending machine” in the context of the provided example).

There is still no semantic error/noise definition, however Bao et al. [3] introduced two kinds of semantic errors based on logic probability:

1. *Unsoundness:* the sent message is true, but the received message is false
2. *Incompleteness:* the sent message is false, but the received message is true

Moreover, some communication tasks might tolerate one type of error more than the other.

Semantic noise refers to the disturbance that affects the understanding of the message, the effect is a semantic mismatch between the transmitter and the receiver.

There is still no rigorous definition of semantic noise, however, it can be categorized in two types:

1. Changes to letters or words in the sentence, such as replacing the synonym of a word, which could lead to alter the ability of the machine to understand the semantic meaning of a sentence. Peng et al. [5] developed a communication system for text transmission to deal with such type of semantic noise.
2. Adversarial semantic noise, that is a kind of subtle noise hardly recognizable by the human eye, which have misleading effects on a machine, an example is provided in *figure 1*.

   Different solutions have been proposed to deal with this kind of noise, Goodfellow et al. [6] proposed a fast gradient sign method to generate perturbation by using the gradient of the loss function, Miyato et al. [7] developed a fast gradient method to generate adversarial examples.

   Training Deep Learning based models with this type of noise is important to increase the robustness of such systems and to prevent possible attacks.

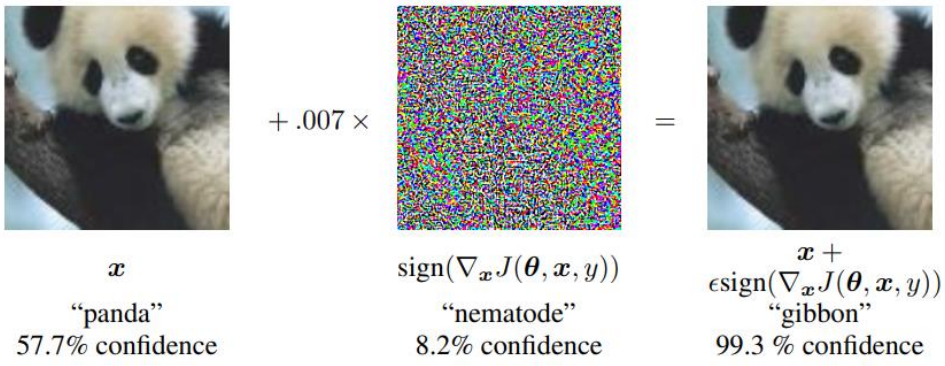


Figure 1: Example of adversarial noise in the image domain ***[6]***

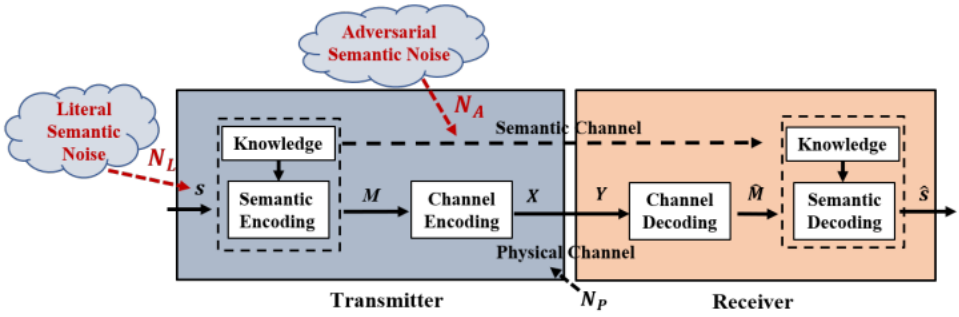


Figure 2: Semantic communication system architecture that considers semantic noise ***[5]***

A noisy semantic channel, in an analogous way with respect to classic information theory channels, has a capacity limit such that a transmission rate can be achieved with relatively small semantic errors. Bao et al. [3] developed a theorem that provides a definition for semantic channel capacity.

Theorem 4: For every discrete memoryless channel, the channel capacity is expressed as:

$$C_s = \sup_{P(Z|X)} \{ I(X;V) - H(Z|X) + \overline{\mathrm{H_s(V)}} \} \tag{11}$$

where $I(X;Y)$ is the mutual information between the source X and the transmission task V, $p(X|W)$ is the is the conditional probabilistic distribution that refers to a

semantic coding strategy with the source, encoded into its semantic representation Z, $H(Z|X)$ is the coding semantic ambiguity and $\overline{\mathrm{H_s(V)}} = -\sum p(v)H(v), v \in V$ is the average logical information of the received messages for the task V.

The channel capacity has the following property: for any $\epsilon > 0$ and $R < C_s$ there is a block strategy such that the maximal probability of semantic error is $< \epsilon$.

Moreover, a high $H(Z|X)$ means high semantic ambiguity, and a high $H_s(V)$ means that the receiver is highly capable of interpreting the received messages, meaning that these two parameters can lead to a channel capacity that could be higher or lower than the Shannon channel capacity, or equivalently Thus, depending on the adopted coding strategy and the receiver ability to interpret the received messages, a semantic communication system can achieve a channel capacity that goes beyond the limit imposed by Shannon Theorem.

Zhijin Qin et al. [8] provided two examples to better understand this concept:

Given the source sentence:" She parked Jame's car on the ground floor of the building, which has 13 floors with 120 sqm on each floor and is called Smith Building due to the creator, William Smith.", the receiver wants to know where Jame's car is.

Case 1: $\overline{H_s(V)} - H(Z|X) > 0$, meaning that the receiver can handle the semantic ambiguity. The source sentence can be compressed as *"the ground floor of smith building"*. Even if the semantic ambiguity increases, the receiver can still complete the task correctly, as a result, C_s is higher than Shannon capacity in this case.

Case 2: $\overline{H_s(V)} - H(Z|X) < 0$, meaning that the receiver cannot handle the semantic ambiguity. The source sentence can be compressed as "She parked Jame's car on the building". the receiver cannot find the car based on the received message and therefore C_s is lower than the Shannon capacity in this case.

1.2.3 Semantic Rate Distortion

In 2021, J. Liu et al. [9] formulated rate distortion as:

$$R(D_s, D_a) = min\, I\left(Z; \hat{X}; \hat{Z}\right) \tag{12}$$

where D_s is the rate distortion between source, X, and recovered information $\hat{X}$, D_a is the rate distortion between the semantic representation Z and the received semantic representation $\hat{Z}$.

The proposed rate-distortion problem seeks a description of the information source, via encoding the extrinsic observation, under two distortion constraints, one for the intrinsic state (corresponding to the semantic aspect of the source) and the other for the extrinsic observation (subject to lossy source coding).

1.2.4 Semantic Information Bottleneck

Information bottleneck is an approach to finding the optimal compromise between compression and accuracy, N. Tishby, F. C. Pereira, and W. Bialek [10] tried to solve the following problem:

$$\min_{P(Z|X)} I(X; Z) - \beta I(V; Z) \tag{13}$$

being V the desired semantic representation.

Definition 5: M. Sana et al. [11] used the previous function as a starting point to develop a new loss function:

$$\mathcal{L} = I(Z; X) - (1 + \alpha) I\left(Z; \hat{Z}\right) + \beta KL\left(X, \hat{Z}\right) \tag{14}$$

where $I(Z; X)$ is the compression term, which represents the average number of bits required for X, $(1 + \alpha) I\left(Z; \hat{Z}\right)$ is the mutual information term, and $\beta KL\left(X, \hat{Z}\right)$ is the inference term, which is the Kullback-Leibler (KL) divergence between the posterior probability at the encoder, X, and the one captured by the decoder $\hat{Z}$, in the end α and β are the parameters to adjust the weight of the mutual information and the inference term.

Though semantic communication systems theory is an evolving research field, it is important to grasp the concepts outlined above, since they could provide valuable insights, particularly in designing such communication systems.

1.3 Performance Metrics

As already mentioned in the previous section, *BER* and *SER* are not suitable to measure semantic communication systems, since the communication focus is shifted from the reliable symbol transmission to the effective semantic information exchange.

In the following of this section, metrics for different types of sources are discussed.

1.3.1 Bilingual Evaluation Understudy (BLEU) score

BLEU score is commonly used to measure the quality of text after machine translation [12], however Xie et al. [13], [14], exploited this metric to measure semantic communication system for text transmission.

Definition 6: The BLEU score between a transmitted sentence s and a received sentence $\hat{s}$ is computed as:

$$logBLEU = min\left(1 - \frac{l_{\hat{s}}}{l_s}, 0\right) + \sum_{n=1}^{N} u_n log P_n \tag{15}$$

where $l_{\hat{s}}$ is the length of the received sequence, l_s is the length of the transmitted sequence, u_n defines the weights of the n-grams, i.e. a contiguous sequence of n items from a given sample, and P_n are the n-grams score, defined as:

$$P_n = \frac{\sum_k min\left(C_k(\hat{s}), C_k(s)\right)}{\sum_k min\left(C_k(\hat{s})\right)} \tag{16}$$

where $C_k(\cdot)$ is the frequency count function for the k-th element in the n-th gram. BLEU score evaluates the difference of n-grams between two sentences, and it ranges from 0 to 1; a higher score indicates greater similarity between two sentences. However, this type of metrics is susceptible to the use of different expressions that yields the same meaning of the substituted word. For instance, the two sentences “my automobile is fast” and “my car is fast” share the same meaning, however the BLEU score will not be one since the length of the n-grams are different.

1.3.2 Sentence similarity

Sentence similarity has been proposed as a solution [13] to the discussed issue. It uses BERT [15], a deep learning model developed by Google, used in the field of natural language processing (NLP) and pre-trained with billions of sentences.

Definition 7: Semantic similarity is computed as:

$$\tau(s, \hat{s}) = \frac{\mathbf{B_\Phi}(\mathbf{s}) \cdot \mathbf{B_\Phi}(\hat{\mathbf{s}})^{\mathbf{T}}}{\left|\left|\mathbf{B_\Phi}(\mathbf{s})\right|\right|\left|\left|\mathbf{B_\Phi}(\hat{\mathbf{s}})\right|\right|} \tag{17}$$

where $\mathbf{B_\Phi}(\cdot)$ is the BERT model, used to map a sentence to its semantic vector space. This means that the two sentences are not directly compared, instead the comparison involves the resulting semantic vectors: the higher the value of τ the higher the similarity.

Additionally, Sana et al. [11] defined a metric to evaluate the trade-off between the transmission accuracy and the number of symbols used for each message:

$$\gamma = \frac{1}{E[n]} \times (1 - \psi(s, \hat{\boldsymbol{s}})) \tag{18}$$

where $E[n]$ is the number of transmitted symbols per message and $\psi(s, \hat{\boldsymbol{s}})$ is the semantic error between the transmitted and the received sentence. Moreover, depending on the task at the receiver it could be computed using different metrics (such as BLEU or MSE).

1.3.3 Image semantic similarity

The commonly used metrics, such as RCNN (PSNR) and structural similarity index (SSIM) fail to count many nuances of human perception, therefore a new metric is required to measure semantic information in image transmissions.

Definition 8: the image semantic similarity [16] between two images is measured as:

$$\eta\big(f(A), f(B)\big) = \left|\left|f(A) - f(B)\right|\right|_2^2 \tag{19}$$

where $f(\cdot)$ is the image embedding function that maps an image point to point in the Euclidean space.

Deep-Learning based image similarity metrics achieve promising results, as convolutional neural networks (CNN) encode high invariance and captures images semantics. Deep CNNs trained on a high-level image classification task are often useful as a representational space.

J.Johnson et al. [17] measure the distance of two images in a Very deep convolutional network (VGG) feature space as the perceptual loss for image regression problems. They define two perceptual loss functions based on a VGG network, ϕ.

Definition 9: The feature reconstruction loss, defined as

$$\mathcal{L}_{feature}^{\phi,l}(A,B) = \frac{1}{L}\left|\left|\phi(A) - \phi(B)\right|\right|_2^2 \tag{20}$$

being $\phi_i(x)$ the activation function of the *l*-th layer, which is of shape L, encourages the two images to have similar representations computed by ϕ.

Definition 10: The style reconstruction loss, defined as

$$\mathcal{L}_{style}^{\phi,l}(A,B) = \left|\left|G_l^{\phi}(A) - G_l^{\phi}(B)\right|\right|_F^2 \tag{21}$$

where $G_l^{\phi}(\cdot)$ is the Gram matrix, penalizes differences in colors, textures and common patterns.

R. Zhang et al. [18] conducted an evaluation of deep features across various architectures and tasks, showing performance improvements over all previously established metrics, and aligning with human perception. Additionally, a deep ranking model introduced by J. Wang et al. [19] examines image similarity relationships using triplets: a query image, a positive image and a negative image. The relative similarity ordering in triplets characterizes the image similarity relationship.

Moreover, several metrics have been developed to assess the similarity between images generated by generative adversarial networks (GANs) [20] and natural images, considering the overall image distribution.

1.3.4 Speech quality measurement

The global semantic information, such as the voice of a speaker, text information and speech delay are required to achieve speech reconstruction.

Metrics such as perceptual evaluation of speech quality (PESQ) [21], short-time objective intelligibility (STOI) [22], and perceptual objective listening quality assessment (POLQA) [23] can be adopted to measure the semantic content of speech signals. Note that, however, the mentioned metrics rely on the perceived quality of the received signal.

Frechet deep speech distance (FDSD) and unconditional kernel deep speech distance (KDSD) are utilized in speech synthesis tasks, to assess the quality of synthetized speech. The features of the speech signals are extracted and fed into an assessment model to measure their similarity.

Definition 11: Being $D \in R^{K,L}$ the extracted features of the original speech samples and $D \in R^{\widehat{K},L}$ the synthesized ones, the FDSD is defined as

$$\Gamma^2 = ||\mu_D - \mu_{\widehat{D}}||\text{^}2 + Tr[\mathbf{\Sigma}_D + \mathbf{\Sigma}_{\widehat{D}} - (\mathbf{\Sigma}_D \mathbf{\Sigma}_{\widehat{D}})^{\frac{1}{2}}] \tag{22}$$

where μ is the average and $\mathbf{\Sigma}$ is the covariance matrix.

Definition 12: being $q(\cdot)$ the kernel function, KDSD [24] is defined as

$$\Delta = \frac{1}{\mathrm{K}(\mathrm{K}-1)} \sum_{1 \le i,j \le K,\ i \ne j} q\left(D_i, \widehat{D}_j\right) + \frac{1}{\widehat{K}\left(\widehat{K}-1\right)} \sum_{1 \le i,j \le K,\ i \ne j} q\left(D_i, \widehat{D}_j\right) + \sum_{i=1}^{K} \sum_{j=1}^{\widehat{K}} q\left(D_i, \widehat{D}_j\right) \tag{23}$$

What emerges from the description provided in this section, is that the appropriate metric to be used for a certain task is heavily dependent on the task itself.

1.4 Challenges

Semantic communications are a new communication paradigm with breakthrough potential; however, this also means that a lot of questions have still to be answered, in the following, a summary of the main open points is discussed.

Semantic theory: although semantic theory has been explored in past decades, most efforts have relied on logical probability, limiting the application scenarios. Moreover, it appears still unsure whether semantic information can be quantified by the concepts of semantic entropy, semantic channel capacity, semantic level rate-distortion theory, and the relationship between, inference accuracy and transmission rate.

Semantic transceiver: designing noise robust semantic communication systems represents a significant challenge. Moreover, a general Joint Source Channel Coding (JSCC) for different information sources is not available yet.

Resource allocation: in contrast with conventional communication systems, which focuses on traditional engineering issues (e.g. improving bit transmission rate), semantic communication systems must address semantic issues too. Resource allocation in this case must have the objective to improve communication efficiency in semantic domain. However, is still unknown how to measure semantic communication efficiency, and how to formulate a resource allocation problem for different task-oriented semantic systems.

Performance metrics: though different metrics have been proposed, there is still no metric to evaluate the amount of semantic information that has been preserved or missed. Moreover, a more general performance metric, such as *BER* or *SER*, is required to evaluate the performance of different semantic communication systems.

Applications: semantic communications applications have still to be determined, this new paradigm has attracted interest from VR/AR applications, as well as from the 6G projects, however, it is not fully clear yet how this technology will be implemented, and which are other possible use cases.

2. State of the art

The following chapter focuses on recent developments in signal processing algorithms, as well as in AI and ML techniques, that enable the real time extraction of semantic information from a given input.

Mert Kalfaa, Mehmetcan Gok et al [25] provide a survey on the latest studies concerning ML techniques, including convolutional and recurrent deep neural networks (DNN) architectures and scene graph generation techniques that make it possible to efficiently extract semantic information from signals of various modality, such as speech, image and video signals.

Moreover, to be able to define suitable performance metrics and to implement compression or coding, the definition of a language that maps meanings to a predefined syntactic structure is needed.

Therefore, it is of critical importance to establish semantic information and language models that are sufficiently general to be suitable in various signal processing applications, keeping in mind that these models must be simple enough to be used by agents with stringent power and computing limitations.

The next sections of this chapter present a review of different semantic language modalities.

2.1 Deep Learning Approaches to Semantic Communications

2.1.1 Natural Languages

NLP is a research area that explores how computers can be used to understand and manipulate natural language text and speech to perform useful tasks [26].

A vast amount of work has been done to generate NL sentences given an input signal, some notable examples are question answering [27], image and video captioning [28], and discourse parsing [29].

More recently, works such as "Semantics-empowered communication for networked intelligent systems" by M. Kountouris, N. Pappas [30], "Universal Semantic Communication" by B. Juba [31] and "Towards a theory of semantic communication" by Bao et al. [3] provide a focus on semantic communications using NL.

NLs provide a universal knowledge for all agents, however, they require to process a massive knowledge base, such as for example the English Language, which makes these approaches unnecessarily complex for IoT sensors and similar machine-type applications.

2.1.2 Propositional Logic

Carnap and Bar-Hillel [2] introduce propositional logic as a semantic language, in this way, the language is encoded in Boolean symbols corresponding to components and primitive properties.

Logical Operators such as AND ($\wedge$), OR ($\vee$), NOT ($\neg$) etc. are used to combine several symbols to form molecular sentences describing a state.

Propositional logic can be tailored for specific applications, avoiding the NLs complexity; however, it is challenging to incorporate numerical attributes such as velocity and position into this kind of language, resulting in a potentially incomplete information about the signals of interest.

2.1.3 Graph-based Languages

Scene graph generations from images and videos [32], knowledge graphs and graph-based question answering [33], and semantic web applications [34] are some of the most popular applications of the graph-based languages.

Graph-based languages are mathematical constructs that can represent components in a signal, as well as their relationships and states. Graph nodes and edges can include additional attributes to convey a more complete description of the underlying scene, providing a complete description for a variety of signals of interest.

However, NLs still represent the backbone of the majority of graph-based language models, and it can lead to an unnecessary complexity for simple machine-type applications.

To solve this issue, Mert Kalfaa, Mehmetcan Gok et al [25] advocate the use of graph-based languages that can be tailored to specific applications of interest with a relatively small knowledge.

2.2 State-of-the-art in semantic transformations

Semantic information is present in various signal modalities, including textual descriptions of images, knowledge graphs derived from paragraphs, and correlation functions of random processes.

Semantic transformation or semantic extraction, i.e. the process of transforming or extracting this semantic information, involves mapping an input modality to a target semantic modality.

Before delving into the proposed language and signal processing framework for semantic communications, the following sections provide a focus on the main semantic

transformations for effective signal processing and goal-oriented semantic communications reported in the Mert Kalfaa, Mehmetcan Gok, et al. [25] survey.

2.2.1 Object Detection and Segmentation

Object detection is a core semantic transformation from the visual domain to the domain of object classes. Convolutional Neural Network (CNN) represent a fundamental component for object detection methods.

R. Girshick et al [35] introduce Recurrent CNN (RCNN), that utilize selective search to propose candidate regions, each processed independently by a CNN and classified using Support Vector Machines (SVMs).

R. Girshick [36] instead of extracting region features separately, performs a single forward pass through a CNN, dividing the resulting feature map into regions using Region-of-Interest (RoI) pooling.

S. Ren, K. He et al. [37] introduce RCNNs with real-time processing capabilities, where object regions are proposed by Region Proposal Networks (RPNs) instead of selective search. J. Redmon, S. Divvala et al. introduce YOLO [38], which has been updated several times since its first release and does not utilize region proposals. Instead, the input image is divided into cells performing inference on a limited number of boxes within each cell.

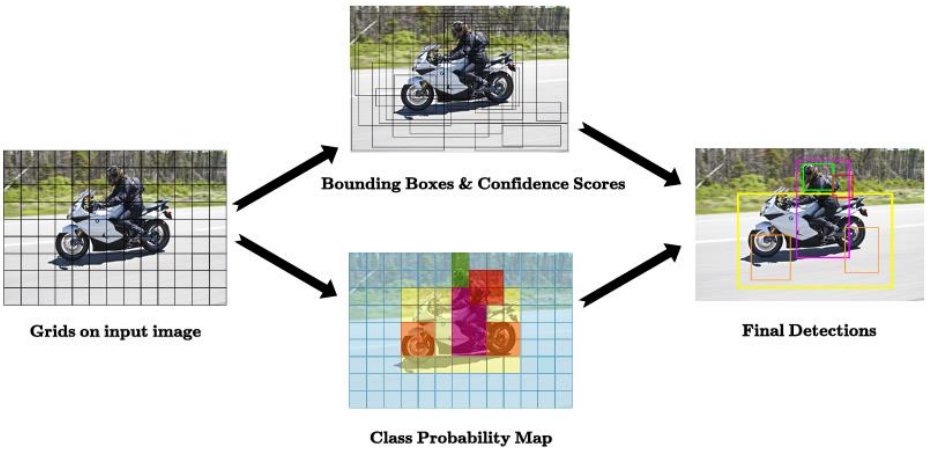


Figure 3: YOLO object detection model

Moreover, recent works [39] [40] employ network scaling approaches to achieve higher detection accuracy within shorter inference times, achieving state-of-the-art results for real-time object detection. M. Tan, R. Pang et al. [40] propose a weighted bi-directional feature pyramid network to fuse features from multiple levels, while [18] applies a network scaling method to the YOLO architecture.

Segmentation is a semantic transformation applicable to visual inputs, which are transformed to 2-D domains such as time-frequency or timescale. Formally, semantic segmentation aims to assign a label from a set of categories to each pixel of the image, treating each pixel as a random variable. Each label can represent an object class, such as a person, plane, or car, or distinct but unspecified clusters in an unsupervised setting.

J. Shotton, M. Johnson and R. Cipolla [41], propose texton forests, which are ensembles of decision trees that act directly on image pixels, as efficient low-level features for image segmentation. Alternative approaches include random forest classifiers [42], and combinations of SVMs and Markov Random Fields (MRFs) [43]. The state-of-the-art in semantic segmentation typically employs convolutional architectures in supervised, semi-supervised, and weakly-supervised settings [44]. The features extracted by the deeper layers of a CNN are more concentrated on concise semantics with low spatial details, whereas shallow layers are more aware of spatial details such as edge orientations.

R. P. Poudel, P. Lamata, G. Montana [45] propose Recurrent Fully Convolutional Networks for multi-slice Magnetic Resonance Imaging (MRI) segmentation, incorporating a Gated Recurrent Unit (GRU) into the bottleneck of the U-Net architecture described in [46]. Furthermore, adversarial training, a protocol in which humans introduce adversarial examples to the model, is applied to semantic segmentation [47].

K. He, G. Gkioxari et al. proposed Mask-CNN [48], a modified version of Faster-RCNN [37], for the instance segmentation task which aims to assign labels to pixels at the object level rather than the class level. Mask-CNN architecture uses ResNet, followed by a Feature Pyramid Network (FPN) and a RPN. Features RoI alignment is

used to extract the proposed regions. Eventually, bounding-box regression, instance classification, and segmentation mask inference are performed.

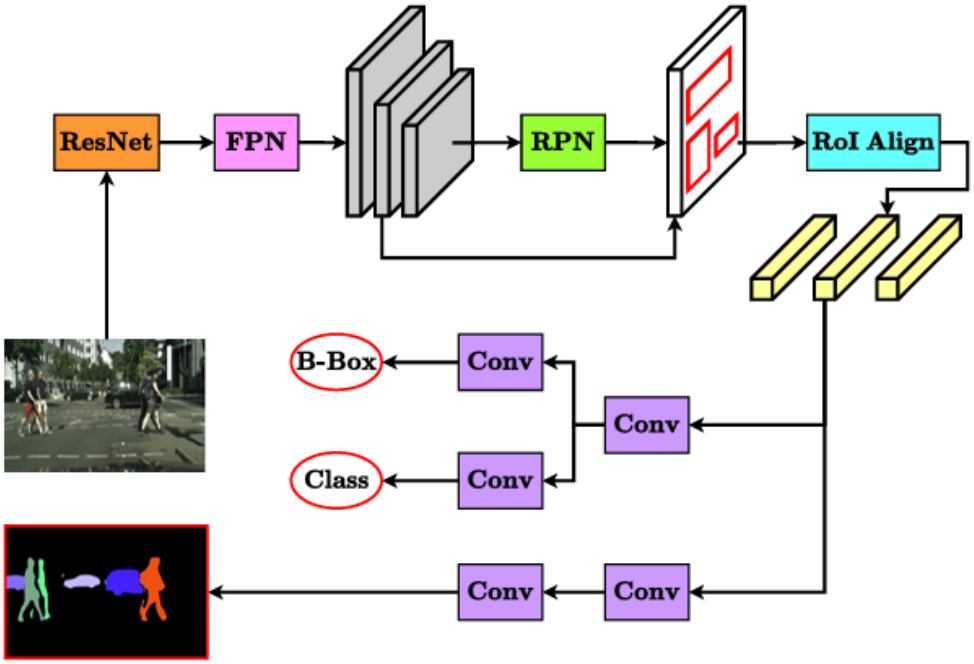


Figure 4: Mask-RCNN architecture

Panoptic segmentation is the union of instance and semantic level segmentation, introduced by A. Kirillov, K. He et al. [49], this method associates each pixel with both instance and class level labels.

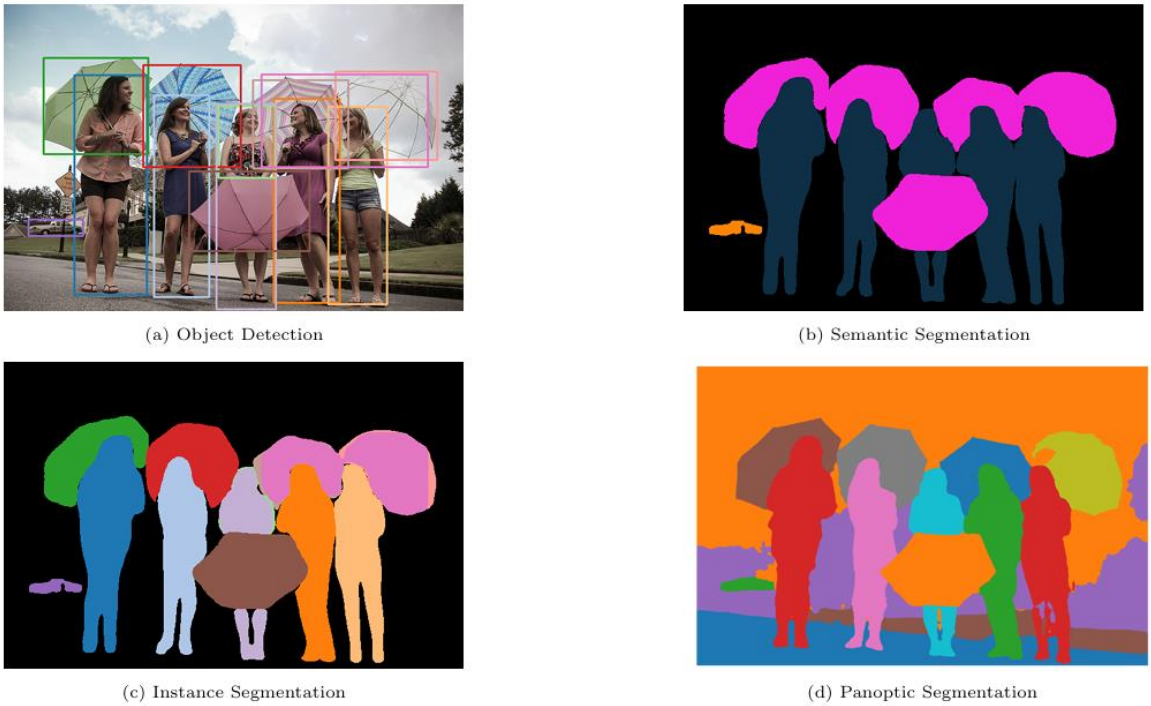


Figure 5: Different object detection and image segmentation techniques applied to the same input

2.2.2 Image and Video Captioning

Image captioning or annotation generation is defined as the process of producing textual descriptions for images; this encompasses not only the identification and description of objects within the frames, but also their relationships and states.

NL descriptions provide an intuitive way to represent the semantic information embedded in image or video. Typically, a CNN backbone is used to extract the visual features from the input signals, and recurrent neural networks (RNN) are then used for sequence modeling as in [50].

Several techniques have been employed to improve the caption quality. A. Karpathy and L. Fei-Fei in [51] propose the captioning on multiple image regions, visual attention on CNNs is instead proposed by K. Xu, J. Ba et al. in [52].

J. Johnson, A. Karpathy and L. Fei-Fei introduce a method where object detection and caption generation tasks are tackled jointly in such a way that the detected visual concepts are described with short NL phrases [53]. However, dense captioning has a

visual concepts localization issue; a possible solution is provided in [54] where global image features are fused with region features.

D.-J. Kim, J. Choi, I. S. Kweon, et al. [55] extend dense captioning task to relational captioning, where given spatial, attentive, and contact relation information, multiple captions are generated for each object pair.

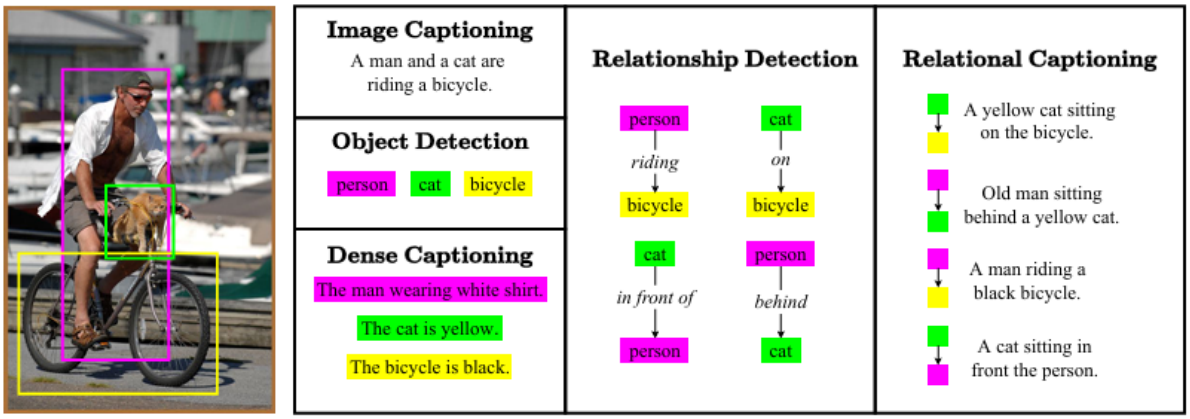


Figure 6: Semantic transformation examples in the image domain

Video captioning, which can be seen as a temporal extension of image captioning, relies on architectures that make use of CNNs (2-D or 3-D) to extract visual semantic content.

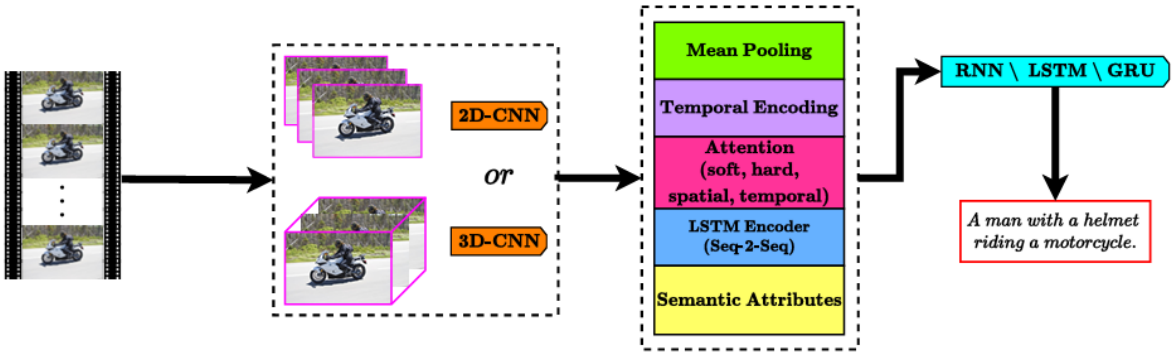


Figure 7: Video captioning pipeline

Once the semantic content is extracted, Long-Short-Term Memory (LSTMs) Networks or RNNs are used to generate NL text sequences, as in [56].

Finally, recurrent visual encoder architectures are employed to extend the applicability of extracted features over longer durations.

2.2.3 Scene Graph Generation

The main idea behind scene graph representation is to convert images into meaningful graphs and encode the visual relationships depicted in the image. The resulting graphical structure is composed by nodes (the detected objects) and edges connecting the nodes (objects relationships).

J. Johnson, R. Krishna et al. [57] propose scene graph-based description of image features (e.g. “man, boat”) and objects relationships (e.g. “man on boat”) and attributes (e.g. “boat is white”). More specifically, the semantic information is retrieved by means of a Conditional Random Field (CRF) model using scene graph queries.

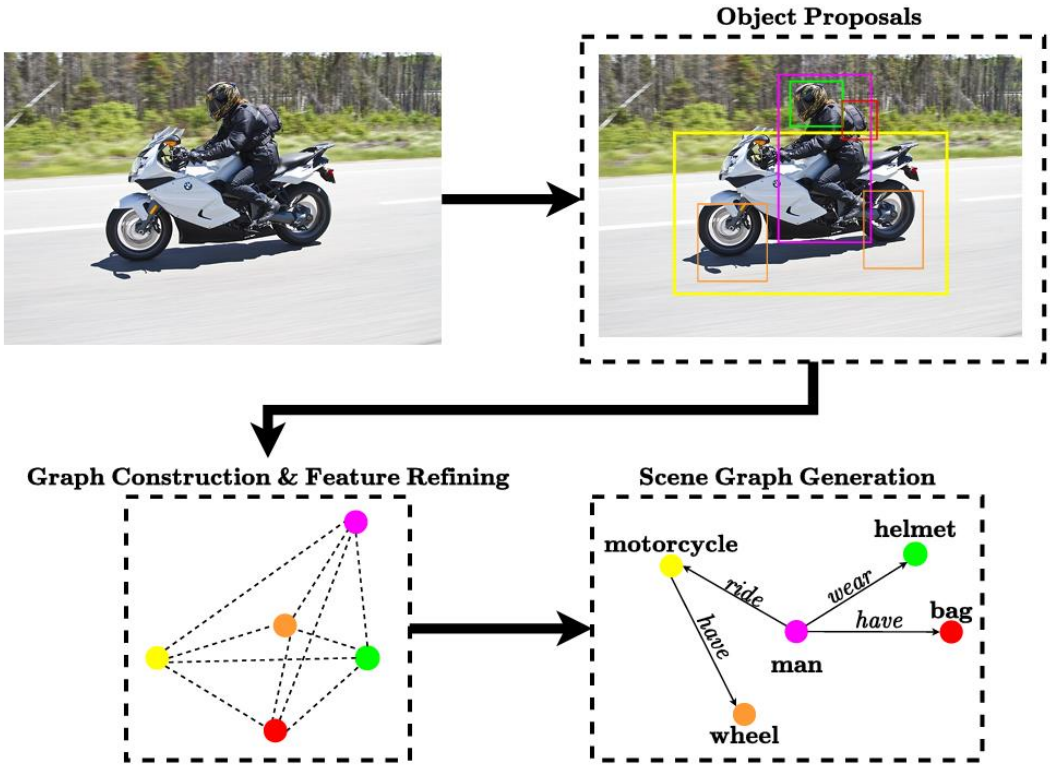


Figure 8: Scene graph generation

More generally, the steps needed to generate the graph representation can be summarized as follows. Given an image, an object detection module (such as Faster-RCNN) extracts object region proposals and their visual features. Identified objects and their extracted features serve as nodes in the initial graph. The features along these nodes and edges are then iteratively refined, and a final graph is inferred based on the refined features.

In [58] a Region Proposal Network (RPN) is used to extract object proposals and proposed objects are paired to obtain a fully connected initial coarse graph.

Graph-RCN [59] prunes the connections in the initial graph using a relation-proposal-network (RePN). Then an Attentional Graph Convolutional Network [60], which operates on graph-structured data, refines the graph features, by leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations.

VCTree model [61] represents another approach to the problem: the features are extracted from objects proposals, which are then used to compute a scoring matrix. Based on this matrix, the model constructs a dynamic tree using reinforce algorithm [62] . Finally, the visual features are encoded into context features and the scene graph is generated using supervised learning.

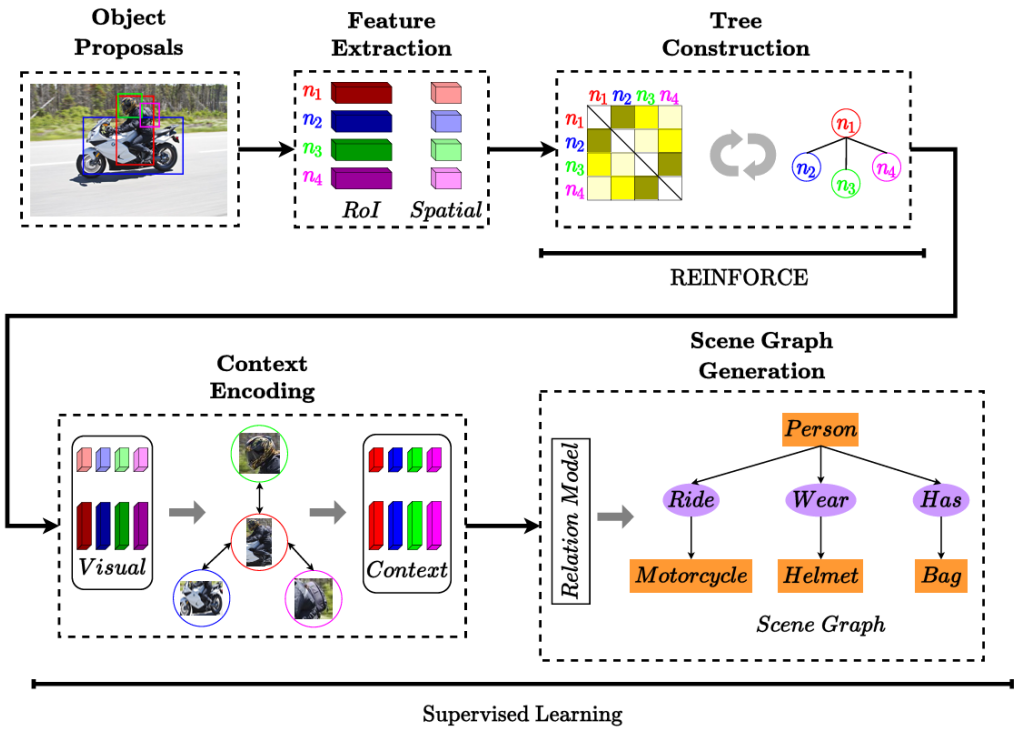


Figure 9: VCTree model

External prior knowledge for scene graph generation tasks is considered as well. In [63] statistics from external texts, such as the conditional probability distribution of a predicate given a (subject, object) pair, are used to regularize visual models. In [64]

natural language priors are incorporated, and visual and textual relationships are jointly learned and aligned, demonstrating that understanding relationships can improve content-based image retrieval.

Scene graphs can be extracted from video signals: in [65] the proposed approach is to convert each frame of a video input signal into a scene graph, using frame and cross-frame relationships to merge semantic information coming from different frames and eventually creating a scene graph that describes the entire video.

2.2.4 Automatic Speech Recognition

M. Malik, M.K. Malik et al. [66] propose a survey about Automatic Speech Recognition (ASR) systems. The conversion of audio signals into NL texts represents, in fact, the most popular application of semantic transformation applied to speech signals.

Figure 10: ASR pipeline

The typical process that an audio signal undergoes during ASR tasks involves an initial preprocessing step (Filtering, DFT, etc.); the audio features (spectral or temporal) are then extracted by means of Mel-frequency Cepstral Coefficients (MFCC) [67], often used for timbral description/comparison, or Discrete Wavelet Transform [68]. The extracted features undergo the prediction phase, that employs Hidden Markov Models (HMM) [69], SVMs [70], RNNs [71] or CNNs [72] to obtain the text equivalent of the audio signal.

2.3 Deep Learning Models for Semantic Communications

This section will provide a comprehensive overview of the state-of-the-art models tailored for different signal inputs, such as video, audio and text.

The analysis of these models will explore how the modern deep learning techniques represent a key player for accurate semantic information extraction/transformation and highlight the critical role of deep learning in advancing this rapidly evolving field.

2.3.1 DeepWiVe: Deep-Learning-Aided Wireless Video Transmission

Video content is the most demanding type of signal in terms of bandwidth. It accounts for 80% of internet traffic, and it is expected to grow further. Therefore, it is necessary to develop sustainable solutions to accommodate this increasing demand. Consequently, more efficient data compression techniques, especially for wireless transmissions, are needed.

Video compression follows the modular approach employed in the conventional transmission systems, where the end-to-end transmission problem is divided into:

- Source encoding problem
- Channel encoding problem

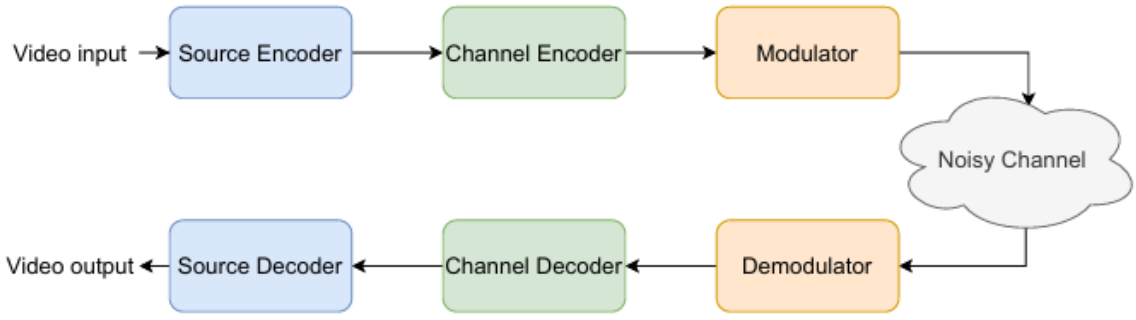


Figure 11: Common separation-based digital video delivery system

This kind of approach, however, starts to show its limits as more challenging video delivery applications emerge, such as virtual reality (VR) or drone-based surveillance systems and in general applications that require ultra-low latency requirements.

Joint source-channel coding (JSCC) represents an alternative to the separation-based architecture: it still uses separate modules for compression and communication, but jointly optimizes their parameters.

DeepWiVe [73] is the first-ever end-to-end JSCC video transmission scheme leveraging the power of DNNs to directly map video signals to channel symbols.

A. Problem formulation

The authors consider the problem of wireless transmission in a constrained bandwidth setting. Consider a video sequence $\boldsymbol{X} = \{\boldsymbol{X}^n\}_{n=1}^{T}$, where $\boldsymbol{X^n} = \boldsymbol{x_1^n}, \dots, \boldsymbol{x_N^n}$, $\boldsymbol{x_i^n} \in R^{HxWx3} \rightarrow C^{Tk}$, $\forall i \in [1, N]$, represents the *n-th* group of pictures (GoP) in the video sequence. Each frame is represented as a 24bit RGB image.

The designed encoding function E maps the video sequence $\boldsymbol{X}$ to a set of complex symbols $z = \boldsymbol{E}(X) \in \boldsymbol{C}^{Tk}$, while the decoding function D maps a noise corrupted version of the encoder output $y = z + n$ to an approximated version of the original sequence $\hat{X} = D(y)$.

In this context the bandwidth restriction is represented by the imposed limitation of k channels per GoP, defining the *bandwidth compression ratio as*:

$$\rho = \frac{k}{3HWN} \tag{24}$$

The additive white Gaussian noise (AWGN) follows a complex gaussian noise distribution with zero mean and covariance $\sigma^2\boldsymbol{I}$, where I is the identity matrix and the average power constraint at the transmitter is

$$\frac{1}{Tk} E_z \left[\left| |z| \right|_2^2 \right] \leq P \tag{25}$$

The channel noise SNR is defined as

$$SNR = 10\log_{10}\left(\frac{\mathrm{P}}{\sigma^2}\right)\mathrm{dB} \tag{26}$$

B. JSCC Model

Consider the *n-th* GoP, the last (x_N^n) is the key frame and is compressed and transmitted by the key frame encoder f_θ

$$z_i^n = f_\theta\left(x_i^n, \widehat{\sigma^2}\right), i = N \tag{27}$$

being $\widehat{\sigma^2}$ the estimated channel noise power at the transmitter.

Each element of z_i^n represents the In-phase (I) and Quadrature (Q) components of a channel complex symbol, which are normalized according to the power and bandwidth constraints.

These values are then transmitted through the channel and the key frame decoder $f_{\theta'}$ maps the noisy received vector back to the original domain

$$\widehat{x_i^n} = f_{\theta'}\left(\hat{y}_i^n, \widehat{\sigma^2}\right), i = N \tag{28}$$

The loss is computed using the peak signal-to-noise ratio (PSNR) [74] or the MS-SSIM [75], which are the quality metrics utilized for this task. The network weights (θ, θ') are then updated via backpropagation with respect to the loss gradient.

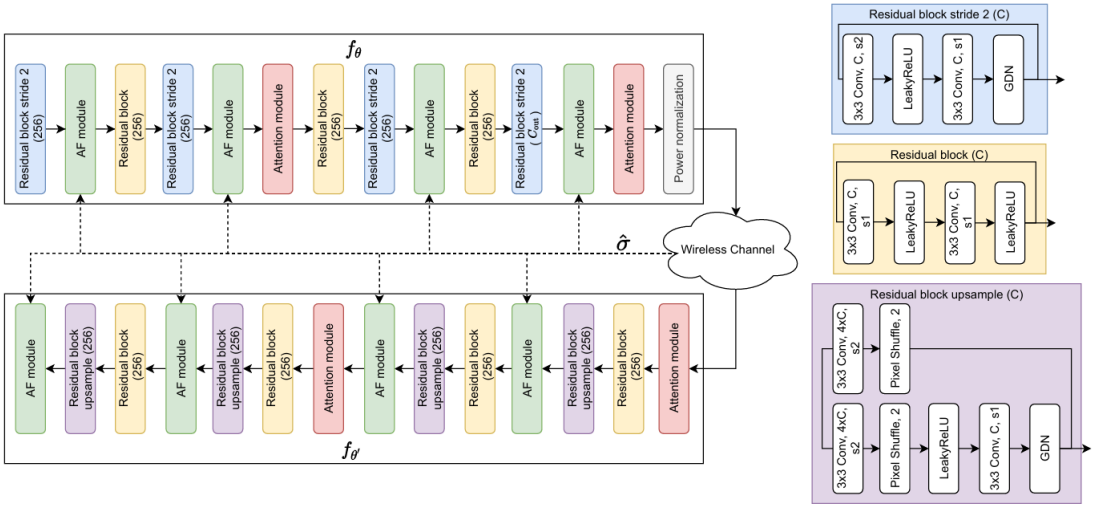


Figure 12: Key frame encoder/decoder network architectures

The remaining frames, i.e. $x_i^n, i = 1,2, \dots, N-1$ the interpolation encoder g_ϕ is used to encode motion information (f_{i-t}^n, f_{i+t}^n) and residual information (r_{i-t}^n, r_{i+t}^n) with respect to two reference frames.

The motion information is estimated by means of a Scaled Space Flow (SSF) [76]: the main idea is to blur regions of the frame where traditional pixel warping struggles to model motion, and instead compensate those regions using the residual information.

To that end, in scale-space warping (SSW) a frame is first transformed into a fixed-resolution volume $\overline{\boldsymbol{X}}_{i+t}^n$,. The SSW $f_{i+t}^n \in R^{HxWx3}$ that warps $\overline{\boldsymbol{x}}_{i+t}^n$ to $\tilde{\mathrm{x}}_{\mathrm{i+t}}^{\mathrm{n}}$ is defined as:

$$\tilde{\mathrm{x}}_{\mathrm{i+t}}^{\mathrm{n}} = SSW\left(\overline{\boldsymbol{x}}_{i+t}^n, f_{i+t}^n\right) \tag{29}$$

The network h_η [77] is used to estimate the space flow f_{i+t}^n, which progressively downsamples the input using convolutional layers before upsampling it back to the original frame dimensions.

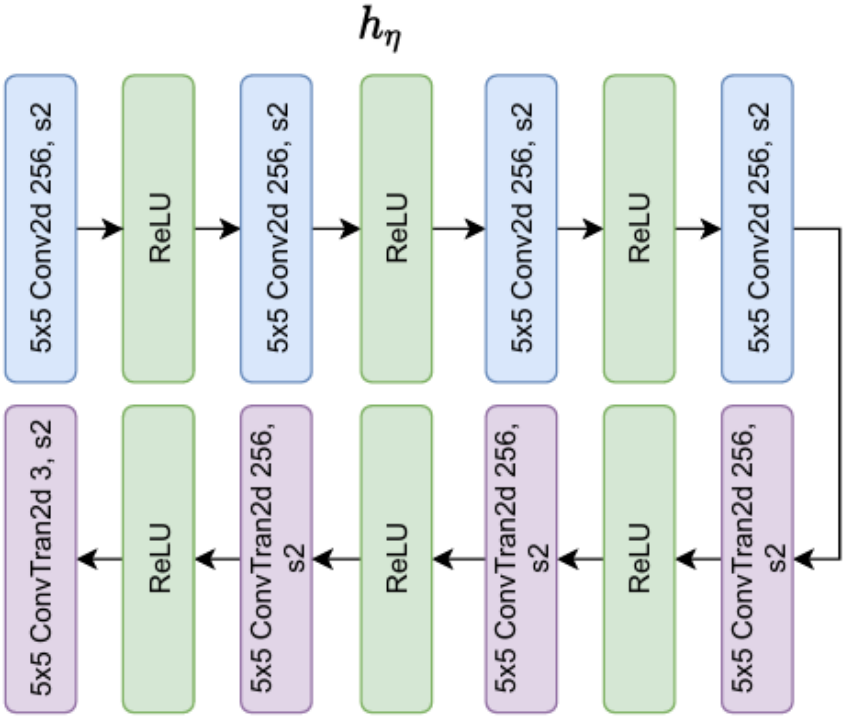


Figure 13: SSF estimator network architecture

The residual is then defined as

$$\mathrm{r}_{\mathrm{i+t}}^{\mathrm{n}} = \mathrm{x}_{\mathrm{i}}^{\mathrm{n}} - \tilde{\mathrm{x}}_{\mathrm{i+t}}^{\mathrm{n}} \tag{30}$$

Then, the interpolation encoder defines the mapping

$$\mathrm{z}_{\mathrm{i}}^{\mathrm{n}} = \mathrm{g}_{\phi}\left(x_i^n, \overline{\boldsymbol{x}}_{i-t}^n, \overline{\boldsymbol{x}}_{i+t}^n, \mathbf{r}_{i-t}^n, \mathbf{r}_{i+t}^n, \mathbf{f}_{i-t}^n, f_{i+t}^n, \widehat{\sigma^2}\right) \tag{31}$$

The z_i^n vector is power normalized and transmitted over the channel, then given $\hat{y}_i^n$, the interpolation decoder estimates the SSF, the residual and a mask.

$$\left(\hat{\mathbf{f}}_{i-t}^n, \hat{f}_{i+t}^n, \mathrm{r}_{\mathrm{i}}^{\mathrm{n}}, \mathrm{m}_{\mathrm{i}}^{\mathrm{n}}\right) = \mathrm{g}_{\phi'}\left(\hat{y}_i^n, \widehat{\sigma^2}\right) \tag{32}$$

where $m_i^n \in R^{HxWx3}$, that, for each H and W index, the sum of values along the channel dimension is equal to 1, which is achieved by softmax activation.

The reconstructed frame is defined as:

$$\hat{\boldsymbol{x}}_i^n = (\mathrm{m}_{\mathrm{i}}^{\mathrm{n}})_1 * SSW\left(\hat{\boldsymbol{x}}_{i-t}^n, \hat{\boldsymbol{f}}_{i-t}^n\right) + (\mathrm{m}_{\mathrm{i}}^{\mathrm{n}})_2 * SSW\left(\hat{\boldsymbol{x}}_{i+t}^n, \hat{\boldsymbol{f}}_{i+t}^n\right) + (\mathrm{m}_{\mathrm{i}}^{\mathrm{n}})_3 * \hat{\boldsymbol{r}}_i^n \tag{33}$$

where $*$ represents element-wise multiplication.

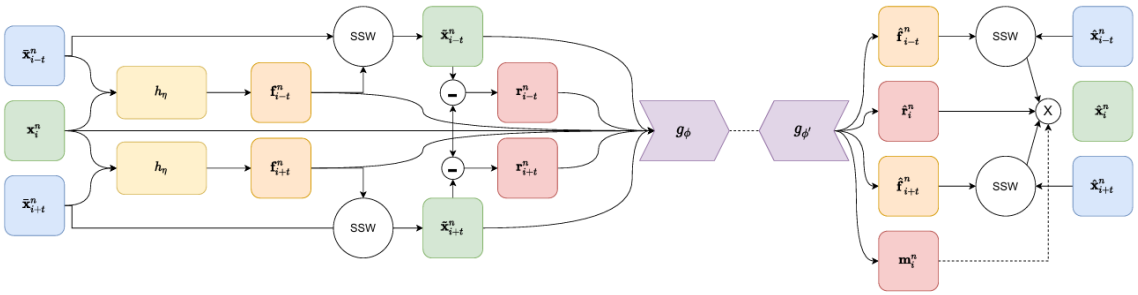


Figure 14: Information flow over the interpolation network

The more diverse are the frames with respect to the reference frames, the more information needs to be transmitted to interpolate the frame accurately. Therefore, for each GoP the available bandwidth must be carefully allocated, the problem is formulated as Markov Decision Process (MDP), defined by the tuple (S, A, P, r), where S is the set of states, A is the action set, P is the probability transition kernel and r is the reward function and solved using reinforcement learning.

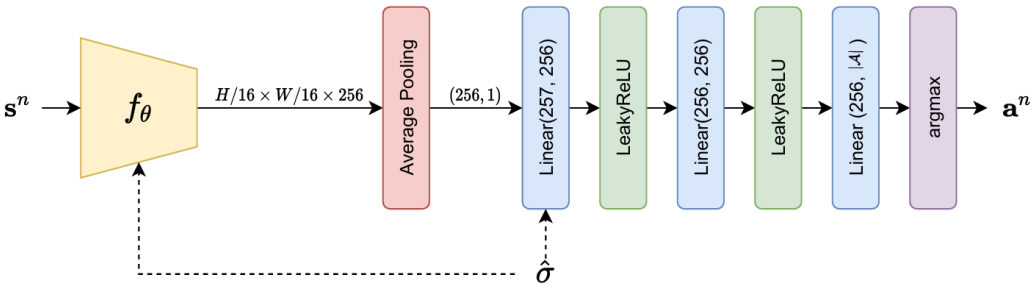
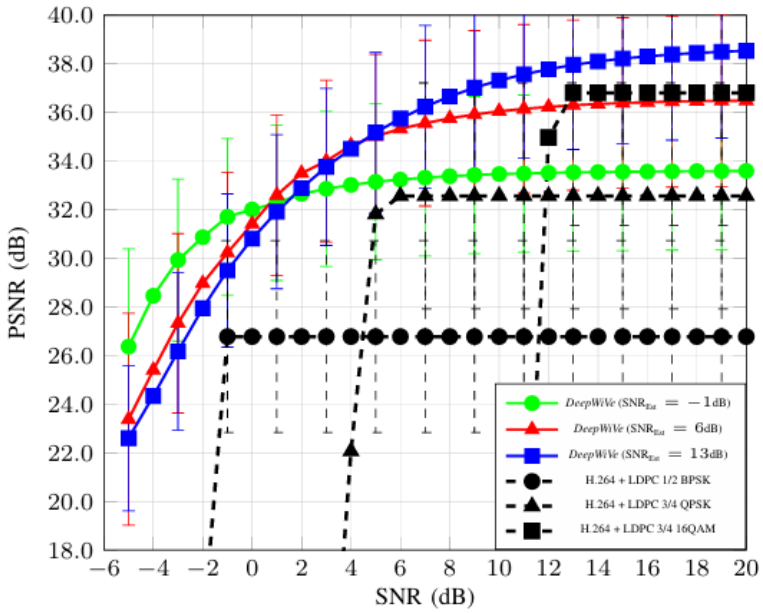


Figure 15: Bandwidth allocation network architecture

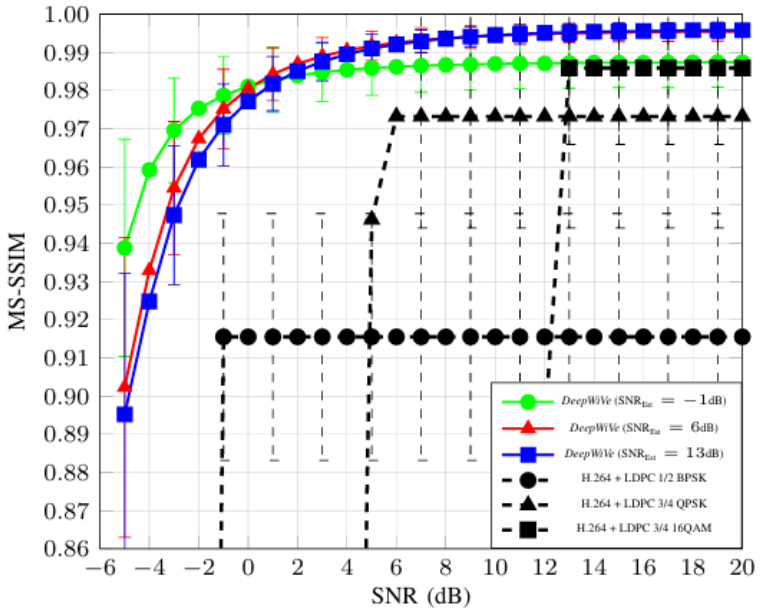
In the end, the DeepWiVe performance are compared to H.264 video compression codecs for source coding, paired with LDPC codes for channel coding.

In Figure 16, the effect of channel estimation error on DeepWiVe performance is displayed: it is possible to observe that the performance of the traditional methods abruptly degrade when a certain SNR_{est} value is reached, which comes from the fact that when the SNR decrease the channel capacity decreases too, leading to certain LDPC rate and modulation order pairs to communicate at a rate higher than the channel

capacity. Instead, it can be seen how DeepWiVe performance degrades gradually for both the considered metrics.



(a) PSNR



(b) MS-SSIM

Figure 16: Effect of channel estimation error on DeepWiVe performance

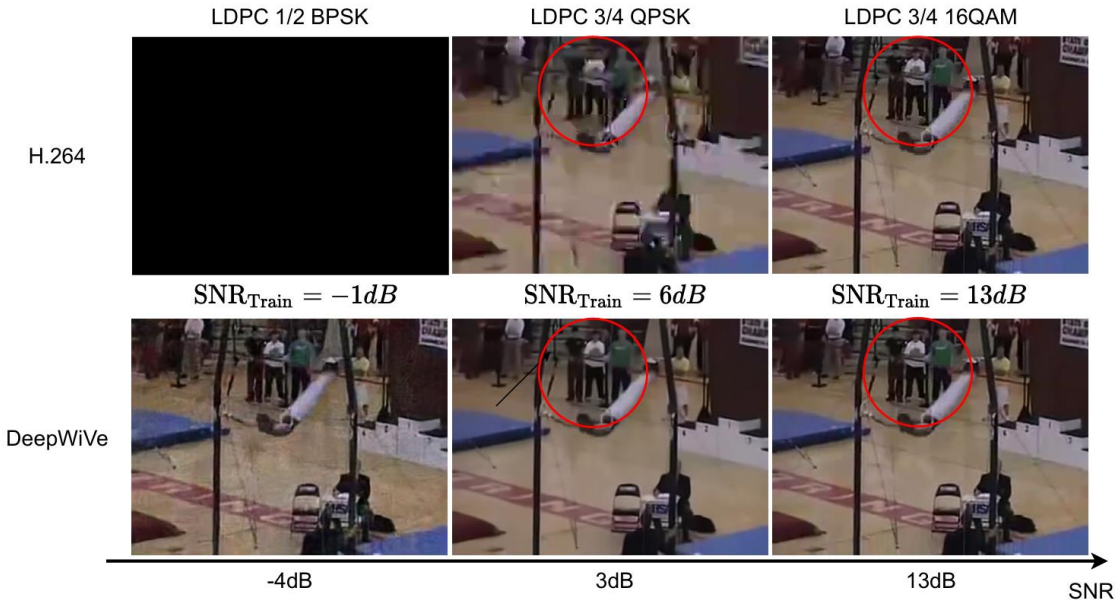
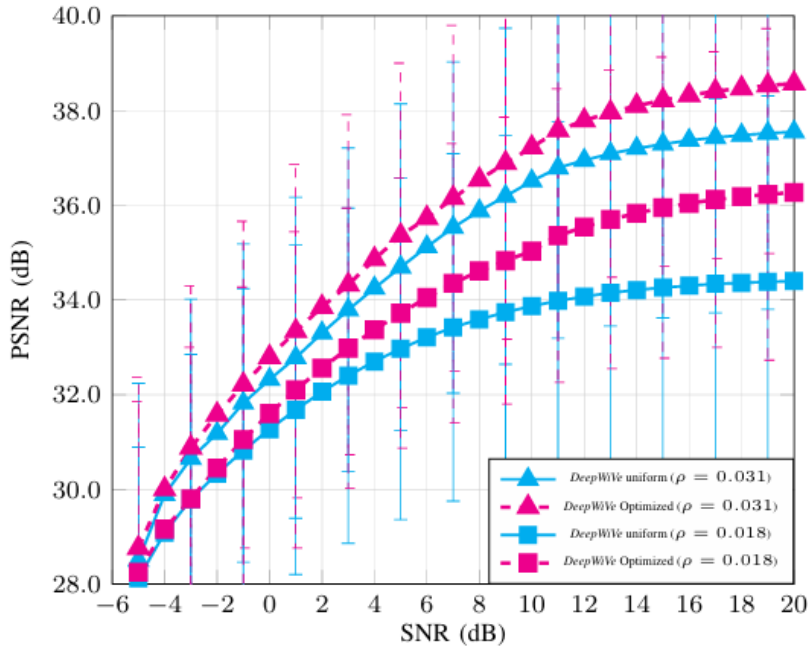
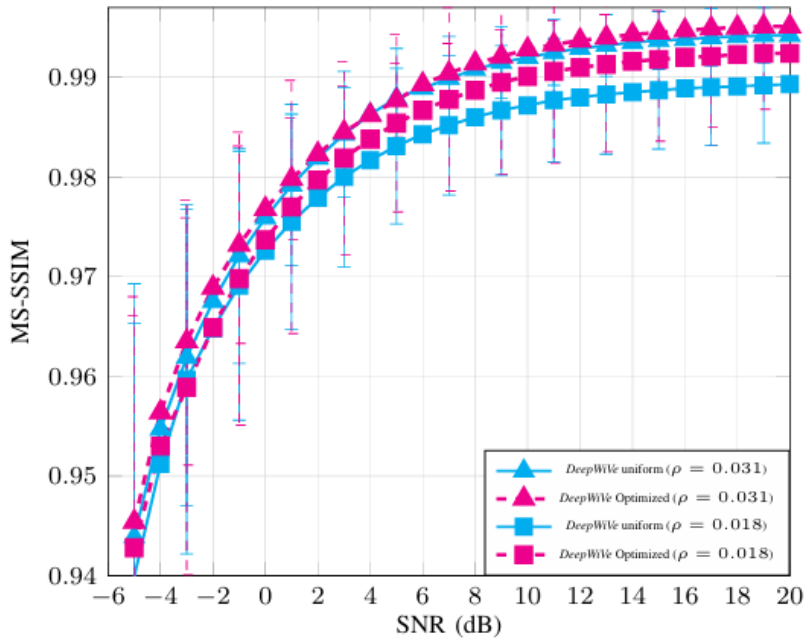


Figure 17: Visual examples of the performance difference

Moreover, the model performance with and without optimal bandwidth allocation is evaluated: Figure **18** shows how the bandwidth allocation significantly improves the system performance. It can also be observed that, when the bandwidth allocation ratio ρ is smaller, i.e., the available channel bandwidth is more limited, the gain from bandwidth allocation is more significant.



(a) PSNR



(b) MS-SSIM

Figure 18: Uniform vs optimal bandwidth allocation comparison

2.3.2 DeepSC

Deep Learning Enabled Semantic Communication System [78], a semantic-based communication system for text, which has been chosen as the starting point for the research activity in this thesis.

The whole system model is developed using the pytorch library [79].

The semantic transceiver consists of:

- a semantic encoder, responsible for semantic information encoding
- a channel encoder, guarantees the correct data transmission over the transmission medium
- a channel decoder, which recovers the received data
- a semantic decoder, responsible for semantic information decoding

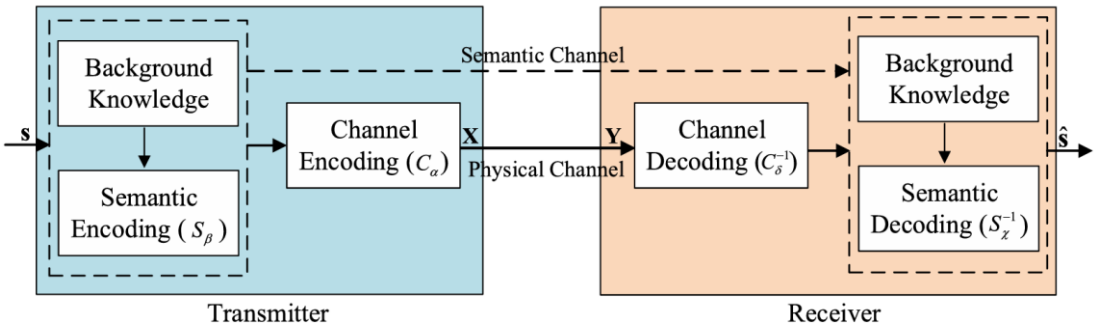


Figure 19: DeepSC framework

The input sentence $\boldsymbol{s} = [w_1, w_2, \dots, w_L]$, where w_i is the l-th word of the sentence, is encoded as

$$\boldsymbol{x} = C_\alpha\left(S_\beta(\boldsymbol{s})\right) \tag{34}$$

where $x \in C^{Mx1}$, with M being the coherent time, $S_\beta(\cdot)$ is the semantic encoder network with the parameter set β and $C_\alpha(\cdot)$ is the channel encoder with the parameter set α.

The received signal, given that x is transmitted, will be

$$\boldsymbol{y} = h\boldsymbol{x} + \boldsymbol{n} \tag{35}$$

The channel allows backpropagation for end-to-end training of the encoder and the decoder, physical channels are formulated by neural networks.

Given the received signal, the decoded sentence can be represented as:

$$\hat{\boldsymbol{s}} = S_{\chi}^{-1}\left(C_{\delta}^{-1}(\boldsymbol{y})\right) \tag{36}$$

$S_{\chi}^{-1}(\cdot)$ is the semantic decoder network with the parameter set χ and $C_{\delta}^{-1}(\cdot)$ is the channel encoder with the parameter set δ.

The cross-entropy (CE) is used as the loss function to measure the difference between the received and the transmitted sequence, and it is formulated as

$$\mathcal{L}_{\mathcal{CE}}(\mathrm{s}, \hat{\mathrm{s}}; \alpha, \beta, \chi, \delta) = \tag{37}$$
$$-\sum q(w_l) log\big(p(w_l)\big) + \big(1 - q(w_l)\big) log\big(1 - p(w_l)\big)$$

where $q(w_l)$ is the real probability that the *l*-th word, w_l, appears in estimated sentence s, and $p(w_l)$ is the predicted probability that the *i*-th word, $\mathrm{w_i}$, appears in sentence $\hat{s}$.

Mutual information is important to provide extra information to train the receiver. It can be computed as follows

$$I(x,y) = \int \chi \times y p(x,y) log \frac{p(x,y)}{p(x)p(y)} dxdy = E_{p(x,y)}\left[log \frac{p(x,y)}{p(x)p(y)}\right] \tag{38}$$

where χ and y are the spaces for x and y, and (x, y) is a pair of random variables from this space. $p(x)$ and $p(y)$ are the marginal probability of sending x and receiving y.

The mutual information is equivalent to the Kullback-Leibler (KL) divergence between the marginal probabilities and the joint probability; therefore, we can re-write the mutual information as:

$$I(x,y) = \mathrm{D}_{KL}(\mathrm{p(x,y)} \,||\, \mathrm{p(x)p(y)}) \tag{39}$$

N. Kalchbrenner et al. [80] state the following theorem:

Theorem 5: The KL divergence admits the following dual representation

$$\mathrm{D}_{KL}(\mathrm{P}||\mathrm{Q}) = \sup_{\mathrm{T}:\Omega\to\mathrm{R}} E_P\,[T] \;-\; log(E_Q\,[e^T]) \tag{40}$$

where the supremum is taken over all functions T such that the two expectations are finite.

Moreover, according to theorem 5, the KL divergence can also be represented as

$$\mathrm{D}_{KL}(\mathrm{p(x,y)}\;||\;\mathrm{p(x)p(y)}) \;\geq\; E_{p(x,y)}\,[T] \;-\; log(E_{p(x)p(y)}\,[e^T]) \tag{41}$$

By utilizing (39) and (41) the mutual information lower bound can be obtained. An unsupervised training method is used for the network T to find a tight bound on the mutual information.

The encoder can be optimized by maximizing the mutual information using the related loss function defined as

$$\mathcal{L}_{MI}(x,y;T) = E_{p(x,y)}[f_T] - log\big(E_{p(x)p(y)}[e^{f_T}]\big) \tag{42}$$

where f_T is composed by a neural network.

The encoder can be optimized by training α and β, i.e. the semantic encoder and the channel encoder parameters, when the mutual information is obtained. Therefore, the loss function can be represented by $\mathcal{L}_{MI}(x,y;T,\alpha,\beta)$

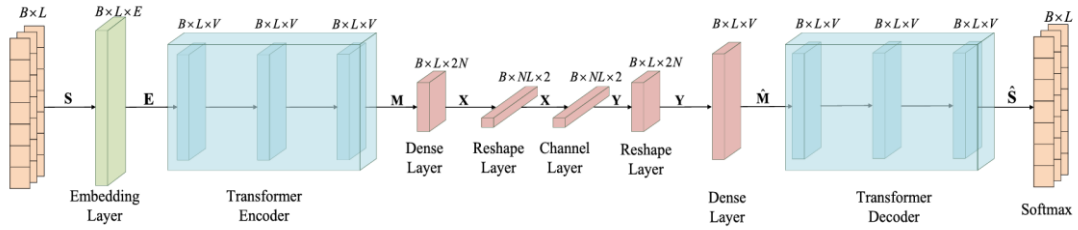


Figure 20. DeepSC network structure

Figure **20** depicts the proposed DeepSC network structure, the input data is processed by an embedding layer and by the semantic encoder, which is composed of multiple transformer encoder layers.

The core of the transformer layer is the multi-head self-attention mechanism, which enables the transformer to view the previous word in the sequence, thereby improving its ability to predict the next word in the sentence.

The output is then processed by the channel encoder, which generates the complex symbols to be transmitted on a channel, which is interpreted as one layer in the model. The original work featured three types of channels, namely, the AWGN channel, the Rician channel and the Rayleigh channel.

At the receiver side the channel decoder is used for symbol detection and, successively, the semantic decoder is used for text estimation.

The loss function used to train the model can be expressed as

$$\mathcal{L}_{total} = \mathcal{L}_{CE}(s, \hat{s}; \alpha, \beta, \chi, \delta) - \lambda \mathcal{L}_{MI}(x, y; T, \alpha, \beta) \quad (43)$$

The first term in (43) is the loss function which evaluates the sentence similarity, that aims to minimize the semantic difference between the transmitted and the received sentence; the second one is the loss function for mutual information, weighted by the parameter λ ($0 \leq \lambda \leq 1$).

Algorithm 1: DeepSC Network Training
Input: The background knowledge set *K,* the initialized weights **W** and bias **b** **Output:** The network $\boldsymbol{S_\beta}(\cdot), \boldsymbol{C_\alpha}(\cdot), \boldsymbol{C_\delta^{-1}}(\cdot), \boldsymbol{S_\chi^{-1}}(\cdot)$ **1:** Create the index to words and words to index dictionaries, and then embedding words. **2: while** stop criterion is not met **do:** **3:** Train the mutual information model **4:** Train the whole network **5: end while**

The training process of the model consists of two phases carried out subsequently, at first, the mutual information model is trained to estimate the achieved data rate, then the whole system is trained with (43) as the loss function. Algorithms 2 and 3 report the pseudo-code of these processes in detail.

Training of Mutual Information Estimation Model: A minibatch is a set of sentences $\boldsymbol{S} \in \mathcal{R}^{BxLx1}$, where B is the batch size and L is the sequences length. The sentences are then represented as a dense word vector $\boldsymbol{E} \in \mathcal{R}^{BxLxE}$, where E is the dimension of the word vector. Subsequently, the semantic information is extracted by the semantic

encoder, obtaining $\boldsymbol{M} \in \mathcal{R}^{BxLxV}$, with V being the dimension of the transformer encoder output.

In order to take into account the physical channel effects, $\boldsymbol{M}$ is encoded into complex symbols $\boldsymbol{X} \in \mathcal{R}^{BxNLx2}$. The received signal $\boldsymbol{Y}$, distorted by the channel noise, is used to compute the mutual information loss $\mathcal{L}_{MI}$. The computed loss is then used to optimize the weights and the bias of $f_T(\cdot)$ through the stochastic gradient descent (SGD).

Training of the whole network: The sample batch is processed in the same way as it is in the first training phase. When $\mathbf{Y}$ is received, the decoded signal $\widehat{\mathbf{M}} \in \mathcal{R}^{\mathrm{BxLxV}}$ is obtained by means of the channel decoder layer. Moreover, the semantic decoder layer estimates the transmitted sentences $\hat{\mathbf{S}}$. Finally, the total loss $\mathcal{L}_{\mathrm{total}}$ is computed and the whole network is optimized by the SGD.

This operation is carried out until the max iteration is met or none of terms in the loss function is decreased any more. Training jointly the two encoders can preserve semantic information when compressing data.

Algorithm 2: Train Mutual Information Estimation Model
Input: The background knowledge set K **Output:** The mutual information estimated model $f_T(\cdot)$ **1: Transmitter:** **2:** BatchSource(K) → **S**. **3:** $\boldsymbol{S_\beta}(\boldsymbol{S})$ → **M**. **4:** $\boldsymbol{C_\alpha}(M)$ → **X**. **5:** Transmit **X** over the channel **6: Receiver:** **7:** Receive **Y** **8:** Compute loss $\mathcal{L}_{MI}$ by (57) **9:** Train T → Gradient descent $(T, \mathcal{L}_{MI})$

Algorithm 3: Train The Whole Network
Input: The background knowledge set K **Output:** The network $\boldsymbol{S_\beta}(\cdot), \boldsymbol{C_\alpha}(\cdot), \boldsymbol{C_\delta^{-1}}(\cdot), \boldsymbol{S_\chi^{-1}}(\cdot)$ **1: Transmitter:**

2: BatchSource(K) → **S**.
3: $\boldsymbol{S_\beta(S)} \rightarrow \mathbf{M}$.
4: $\boldsymbol{C_\alpha}(M) \rightarrow \mathbf{X}$.
5: Transmit **X** over the channel
6: Receiver:
7: Receive **Y**
8: $\boldsymbol{C_\delta^{-1}(Y)} \rightarrow \boldsymbol{\widehat{M}}$.
9: $\boldsymbol{S_\chi^{-1}(\widehat{M})} \rightarrow \boldsymbol{\widehat{S}}$.
10: Compute loss function $\mathcal{L}_{total}$ by (58)
11: Train $\beta, \alpha, \delta, \chi \rightarrow$ Gradient descent $(\beta, \alpha, \delta, \chi, \mathcal{L}_{total})$

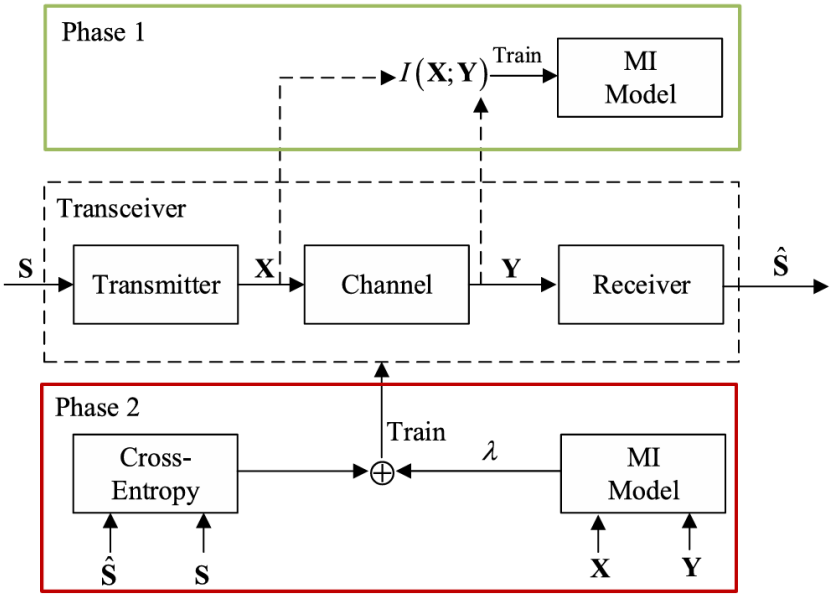


Figure 21: Network training representation: phase 1 trains the mutual information estimation model; phase 2 trains the whole network ***[78]***

2.3.3 L-DeepSC: A Lite Distributed Semantic Communication System for Internet of Things

Internet of Things (IoT) networks are providing more and more intelligent services by processing a massive amount of data [81]. The DL-enabled IoT devices are capable of exploiting different data types; however, the limited capabilities in terms of storage, computing and battery still prevent from wide applications of DL [82], which is usually

trained and updated at the cloud/edge platform based on data from the IoT devices [83].

Therefore, transmitting accurate data to the cloud/edge platform over wireless channels with limited bandwidth and reducing the number of DL parameters to lower the latency and the power consumption represent two crucial problems.

A promising approach to the first problem is represented by semantic communication systems, which are more robust to channel variation and are able to achieve better performance in terms of source recovery, especially in low SNR regime. To deal with the second issue, a network slimmer can be applied to compress DL models without degrading their performance [84].

L-DeepSC [85] is a model developed for IoT devices, starting from the already existing DeepSC [78], to which a network slimmer has been applied. The model focuses on text data, which can be used to generate semantic features to be transmitted to the center to perform intelligent tasks.

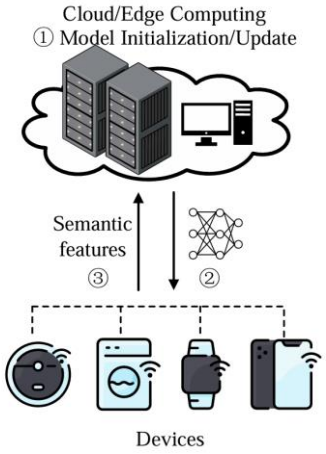


Figure 22: Proposed distributed semantic network

As already stated, DeepSC will be extensively described in the next chapter; however, to understand the following part of this subsection, it is important to grasp the model structure, which can be divided into three parts:

- Transmitter part: which includes semantic encoder and channel encoder

- Physical Channel
- Receiver part: which includes channel decoder and semantic decoder

The main limitations of DeepSC for IoT networks are represented by the huge number of parameters and by the fading channel effects on model training.

The source message *s* is embedded into $\boldsymbol{S}$, and then encoded into:

$$\mathbf{X} = \sigma(\mathbf{W}_{\mathrm{T}}\mathbf{S} + \mathbf{b}_{\mathrm{T}}) \tag{44}$$

where $\boldsymbol{X}$ are the semantic features transmitted to the cloud/edge platform, $\boldsymbol{W}_T$ and $\mathbf{b}_{\mathrm{T}}$ are the trainable parameters and σ is the sigmoid activation function.

From the received symbol, which is affected by the channel matrix and the AWGN, the cloud/edge platform recovers the embedding matrix

$$\hat{\mathrm{S}} = \sigma(\mathbf{W}_R\mathbf{Y} + \mathbf{b}_R) \tag{45}$$

After the de-embedding layer, the estimated source message $\hat{s}$ is retrieved and the parameters can learn to recover the original message *s*. The model uses the same loss function used in DeepSC for optimization purposes.

The channel impacts can be mitigated by exploiting CSI at the cloud/edge, if $\boldsymbol{H}$ is known then the received symbol can be processed by

$$\widetilde{\mathbf{Y}} = (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{Y} = \mathbf{X} + \widetilde{\mathbf{N}} \tag{46}$$

where $\widetilde{\mathbf{N}} = (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{N}$. The previous discussion shows the importance of CSI. Genearlly, CSI can only be estimated generally by means of traditional estimators, such as least-squared (LS), linear minimum mean-squared error (LMMSE) or minimum mean-squared error (MMSE). The authors use LS estimator for simplicity.

To increase the LS estimator resolution, a deep de-noise network, i.e. attention-guided denoising convolutional neural network (ADNet) [86], is exploited. ADNet includes four blocks:

1. Sparse block: extracts features from the input
2. Feature enhancement block
3. Attention block: extracts hidden noise information
4. Reconstruction block: reconstructs the de-noised image

The rough CSI estimated by the LS estimator with few pilots is

$$\boldsymbol{H}_{rough} = \mathbf{Y}_{\mathrm{p}}\mathbf{X}_{\mathrm{p}}^{\mathrm{H}} = \mathbf{H} + \mathbf{N}\mathbf{X}_{\mathrm{p}}^{\mathrm{H}} = \boldsymbol{H} + \widehat{\boldsymbol{N}} \tag{47}$$

where $\boldsymbol{Y}_p = \boldsymbol{H}\boldsymbol{X}_P + \boldsymbol{N}$ is the receiver pilot signal, $\boldsymbol{X}_p$ is the transmitted pilot signal and $\widehat{\boldsymbol{N}} = \boldsymbol{N}\boldsymbol{X}_p^H$.

The refined CSI is denoted as

$$\boldsymbol{H}_{refine} = \mathrm{ADNet}\left(\mathbf{H}_{\mathrm{rough}}\right) \tag{48}$$

ADNet is trained using the loss function $\mathcal{L}\left(\boldsymbol{H}_{refine}, \boldsymbol{H}\right) = \frac{1}{2}||\boldsymbol{H}_{refine} - \boldsymbol{H}||_F^2$, with proper training ADNet can mitigate noise impacts without prior channel information.

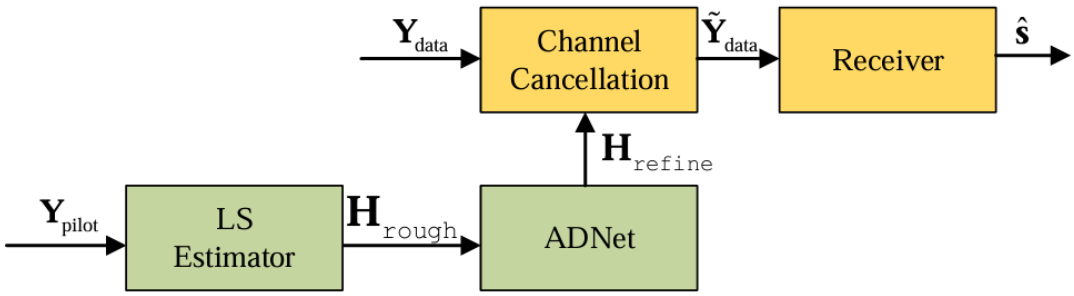


Figure 23: Proposed CSI refinement

To reduce latency due to the huge amount of model weights, the proposed solution is a pruning-quantization mixture. Initially, weights are pruned according to the following algorithm:

Algorithm 4: Network Sparsification
Input: The pre-trained weights **W**, the sparse ratio γ. **Output:** The pruned weights $\boldsymbol{W}_{pruned}$ **1:** Count the total number of connections, M. **2:** Sort the whole connections from small to large, s. **3:** Obtain the threshold by (37) with *M* and γ, w_{thre} **4: for** n = 1 to *N* **do** **5:** Prune the connections by (36), $\boldsymbol{W}_{pruned}^{n}$ **6: end for** **7:** Fine-tune the pruned model by loss function (35).

The loss function is

$$\mathcal{L}_{CE}(s,\hat{s}) = \sum_{i=1}(q(w_i)-1)log\big(1-p(w_i)\big) - \sum_{i=1} q(w_i)log\big(p(w_i)\big) \tag{49}$$

where $q(w_i)$ is the probability that the word w_i appears in the sentence *s*, and $p(w_i)$ is the predicted probability that the word w_i appears in the reconstructed sentence $\hat{s}$.

The pruning function is:

$$W_{i,j}^{n} = \begin{cases} W_{i,j}^{n}, & if\left|W_{i,j}\right| > w_{thre}, \\ 0, & otherwise \end{cases} \tag{50}$$

with

$$w_{thre} = s_{Mx\gamma} \tag{51}$$

where *s* is the sorted weights value from the least to the most important one, *M* is the total number of connections, and γ is the sparsity ratio between 0 and 1, which indicates the proportion of values to prune.

The quantization step follows instead the following algorithm:

Algorithm 5: Network Quantization

Input: The pre-trained weights **W**, the quantization level m, the correlation coefficient c, and the calibration data K.
Output: The pre-trained weights $\boldsymbol{W}_{quantized}$ and the activation range $[x_{min}, x_{max}]$
1: Phase 1: Weights Quantization.
2: for n = 1 to N **do**
3: Compute the range of weights, $[min(W^n), max(W^n)]$
4: Quantize the weights by (38), $\widetilde{\boldsymbol{W}}^n$
5: end for

6: Phase 2: Activations Quantization
7: for k = 1 to K **do**
8: **for** n = 1 to N **do**
9: Update the dynamic range of activation by (40) and (41), $[x_{min}^n(t), x_{max}^n(t)]$
10: **end for**
11: end for
12: Quantize the activations by (42).
13: Fine-tune the quantized model by STE and loss function

The quantization function is

$$\widetilde{\boldsymbol{W}_{i,j}}^{n} = round\left(q_w\left(\boldsymbol{W}_{i,j}^n - min(\boldsymbol{W}^n)\right)\right) \tag{52}$$

where q_w is the scale factor to map the dynamic range of float points to an m-bits integer, given by

$$q_w = \frac{2^m - 1}{max(\boldsymbol{W}^n) - min(\boldsymbol{W}^n)} \tag{53}$$

An exponential moving average (EMA) is used to reduce the influence from the outliers, the range is computed as

$$x_{min}^n(t+1) = (1-c)x_{min}^n(t) + cmin\left(\boldsymbol{X}^n(t)\right) \tag{54}$$

$$x_{max}^n(t+1) = (1-c)x_{max}^n(t) + cmax\left(\boldsymbol{X}^n(t)\right) \tag{55}$$

The activations output is quantized by

$$\widetilde{\boldsymbol{X}^n} = clamp\left(round\left(q_x(\boldsymbol{X}^n - x_{min}^n)\right); -M, M\right) \tag{56}$$

$clamp(\cdot)$ is used to eliminate the quantized outliers, and it is deifned as

$$\mathrm{clamp}(\mathrm{X^n}; -\mathrm{T}, \mathrm{T}) = min(max(X^n, -T), T) \tag{57}$$

where $T = 2^m - 1$, i.e. the border of the m-bits integer format.

The straight-through estimator (STE) is used to estimate the gradient of the quantized weights in the backpropagation; this is necessary since the rounding operation is not derivable.

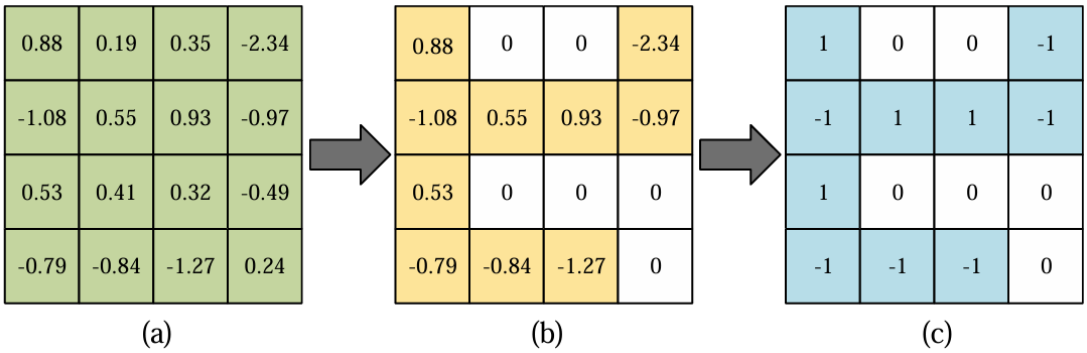


Figure 24: Flowchart of the proposed joint pruning-quantization, the values serve as an example.
(a) shows the original weight matrix, (b) the pruned weights, (c) the quantized weights

The constellation resulting from the semantic source message (Figure **25**) is more complex with respect to the traditional bits constellations: since it is not limited to few points, it is more demanding from the hardware point of view. Therefore, the two-stage quantization process is used to narrow the range of constellations. In fact, the learned high-resolution constellation is converted into a low-resolution constellation with fewer points.

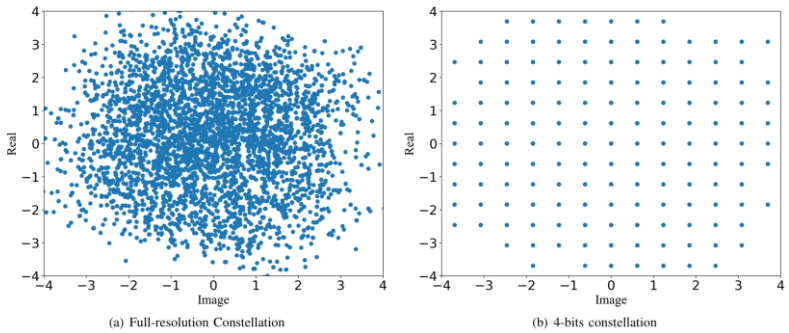


Figure 25: Comparison between full-resolution constellation and low-resolution constellation

The performance of the same constellation with different resolutions is tested, Figure **26** shows that even a 4-bits constellation achieves good performance in terms of BLEU score

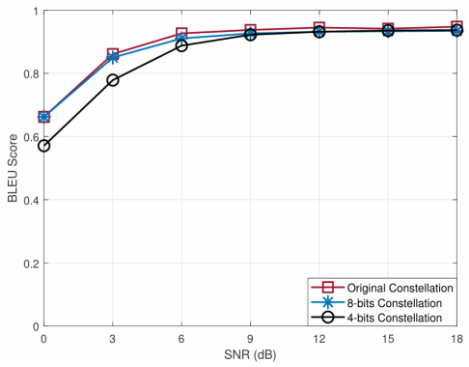


Figure 26: BLEU scores for different constellation sizes

Moreover, the performance of different estimators is compared, showing how, in terms of Mean Squared Error (MSE), the LS estimator with ADNet outperforms the other considered estimators (Figure **27**).

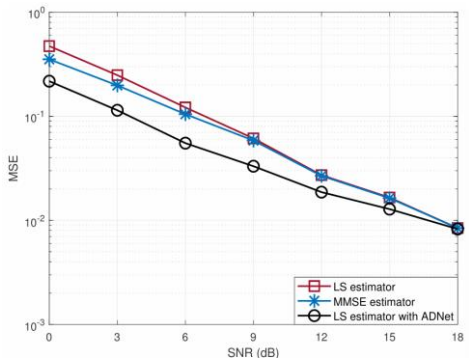


Figure 27: MSE for different types of estimators

The next comparison considers L-DeepSC versus more traditional approaches over Rayleigh and Rician channels for a certain range of SNR values, showing that L-DeepSC outperforms the mentioned approaches.

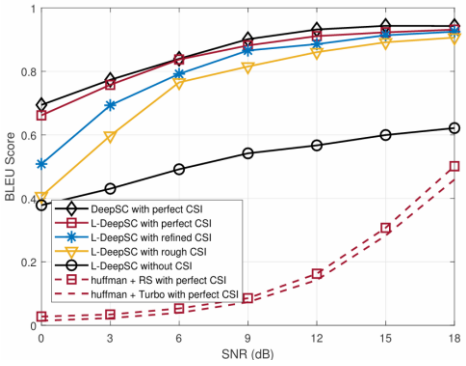


Figure 28: BLEU score vs SNR under Rician fading channel

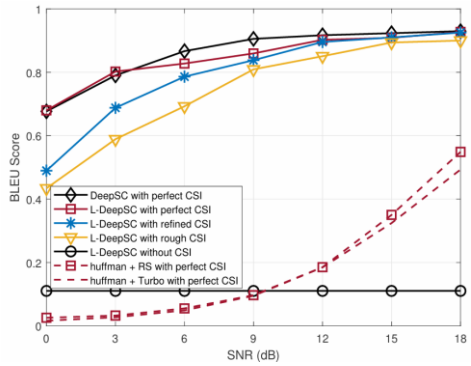


Figure 29: BLEU score vs SNR under Rayleigh fading channel

Then the influence of the sparsity ratio γ and the quantization ratio *m* on the model performance is assessed, showing how the performance over different SNRs is almost unaffected until a limit value, which is > 0.9 for γ and < 4 for m, is reached.

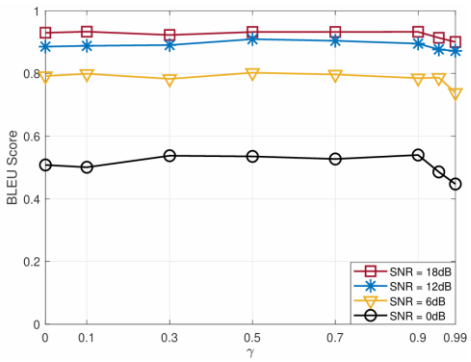


Figure 30: BLEU scores of different SNRs versus sparsity ratio $\boldsymbol{\gamma}$

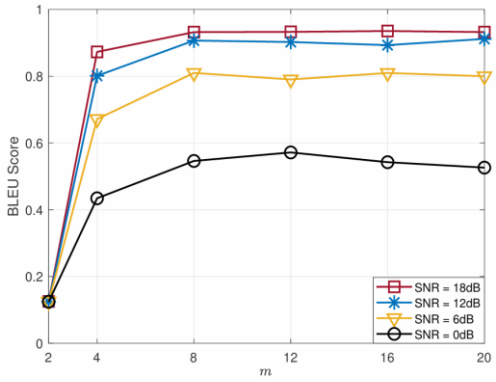


Figure 31: BLEU scores of different SNRs versus sparsity ratio m

Note that the authors do not specify under which conditions the Huffman + Rs with perfect CSI and the Huffman + Turbo with perfect CSI are tested. For this reason, even if the proposed model shows promising results, further tests are preferable to confidently state that L-DeepSC offers unequivocally better performance than the traditional networks.

2.3.4 DeepSC-S: Semantic Communication System for Speech Transmission

Most DL-based pre-processing techniques for speech signals focus on the magnitude, spectra, or Mel-frequency Cepstrum [87], before feeding into a learning system. These extra operations capture the unique features of speech signals; however, they run counter to the motivations behind artificial intelligence.

This is the motivation that led Zhenzi Weng and Zhijin Qin to propose a DL-enabled semantic communication system for speech signals, named DeepSC-S [88].

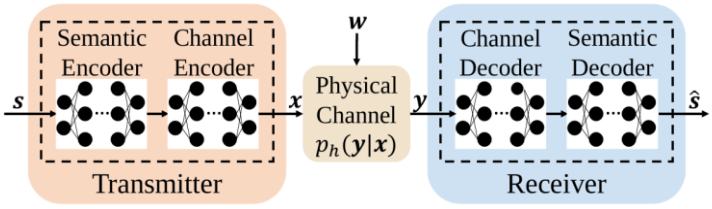


Figure 32: The proposed system model [88]

The system input $\boldsymbol{s} = [s_1, s_2, \dots, s_W]$ is a sample sequence with W samples, drawn from a speech dataset. A batch of input sequences $\boldsymbol{S} \in \boldsymbol{R}^{BxW}$, where B is the batch size, is fed each time to the transmitter. The input sample sequences, are framed into $\boldsymbol{m} \in$

$\boldsymbol{R}^{BxFxL}$ for training before passing through an attention-based encoder, i.e. the semantic encoder $\boldsymbol{T}_{\alpha}^{S}$, where F is the number of frames and L is the length of each frame. The semantic encoder outputs the learned features $\boldsymbol{b} \in \boldsymbol{R}^{BxFxLxD}$. The channel encoder, $\boldsymbol{T}_{\beta}^{C}$, which is a CNN layer with 2D CNN modules, converts $\boldsymbol{b}$ into $\boldsymbol{U} \in \boldsymbol{R}^{BxFx2N}$. Finally, $\mathbf{U}$ is reshaped into the encoded symbol sequence $\mathbf{X} \in \mathbf{R}^{\mathrm{BxFNx2}}$in order to be transmitted as complex symbols. Note that the NN parameters of the semantic encoder and the channel encoder are denoted as $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively.

Thus, each encoded symbol sequence is expressed as:

$$\boldsymbol{X} = \boldsymbol{T}_{\beta}^{C}\left(\boldsymbol{T}_{\alpha}^{S}(\boldsymbol{S})\right) \tag{58}$$

The transmitted symbols are normalized to ensure that the total transmitted power is equal to 1.

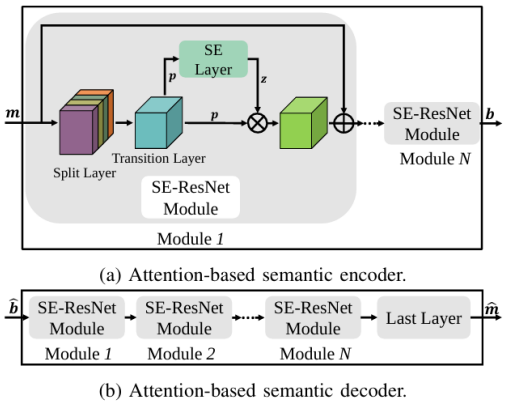


Figure 33: The proposed semantic encoder and semantic decoder structures [88]

The channel layer takes $\boldsymbol{x}$ as input and outputs the signal $\boldsymbol{y}$, modeled as

$$\boldsymbol{Y} = \boldsymbol{H} * \boldsymbol{X} + \boldsymbol{W} \tag{59}$$

Where $\boldsymbol{H}$ are the coefficients of a linear channel and $\boldsymbol{W}$ indicates independent and identically distributed (i.i.d) Gaussian noise samples.

In a similar way with respect to the transmitter, the receiver consists of a channel decoder, $\boldsymbol{R}_{\chi}^{C}$, to mitigate channel distortion and attenuation, and a semantic decoder, $\boldsymbol{R}_{\delta}^{C}$, recovers speech signals based on the extracted, and learned, semantic features.

Note that χ and δ represent the channel and semantic decoder parameters. The decoded signal $\hat{s}$ is obtained by

$$\widehat{\boldsymbol{S}} = \boldsymbol{R}_{\delta}^{C}\left(\boldsymbol{R}_{\chi}^{C}(\boldsymbol{Y})\right) \tag{60}$$

The MSE is used as the loss function to measure the difference between $\boldsymbol{S}$ and $\widehat{\boldsymbol{S}}$

$$\mathcal{L}_{MSE}(\boldsymbol{\theta}^{T}, \boldsymbol{\theta}^{R}) = \frac{1}{W}\sum_{w=1}^{W}(s_w - \widehat{s_w})^2 \tag{61}$$

being $\boldsymbol{\theta}^{T}$ and $\boldsymbol{\theta}^{R}$ the transmitter and the receiver parameters respectively, and s_w and $\widehat{s_w}$ the w-th element of vectors $\boldsymbol{s}$ and $\hat{\boldsymbol{s}}$ respectively.

Note that semantic encoder/decoder and channel encoder/decoder are jointly designed; therefore, both parameter sets $\boldsymbol{\theta}^{T}$ and $\boldsymbol{\theta}^{R}$ can be adjusted at the same time. More specifically, the SGD algorithm is adopted for training; denoting the parameters of the whole system as $\boldsymbol{\theta}$, the update is iteratively carried out as follows:

$$\boldsymbol{\theta}^{(i+1)} \leftarrow \boldsymbol{\theta}^{(i)} - \boldsymbol{\eta}\nabla_{\boldsymbol{\theta}^{(i)}}\mathcal{L}_{MSE}(\boldsymbol{\theta}^{T}, \boldsymbol{\theta}^{R}) \tag{62}$$

where η is a learning rate and ∇ is the differential operator.

The metrics employed to evaluate the system performance are:

- the signal-to-distortion ratio (SDR) [89]

$$SDR = 10log_{10}\left(\frac{||\boldsymbol{s}||^2}{||\boldsymbol{s} - \hat{\boldsymbol{s}}||^2}\right) \tag{63}$$

  which represents that the speech information is recovered with better quality, i.e. easier to understand for human beings

- The perceptual evaluation of speech distortion (PESQ) [90], integrated by means of an open-source assessment model developed by ITU-T [91].

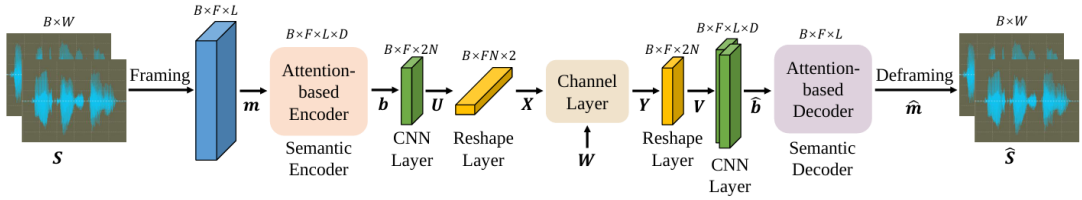


Figure 34: The proposed system architecture [88]

First, DeepSC-S performance in terms of MSE under AWGN, Rayleigh and Rician channels are evaluated, it can be seen how the model performs poorly with Rayleigh channels, however, the MSE values achieved under the Rician channel condition makes the model robust.

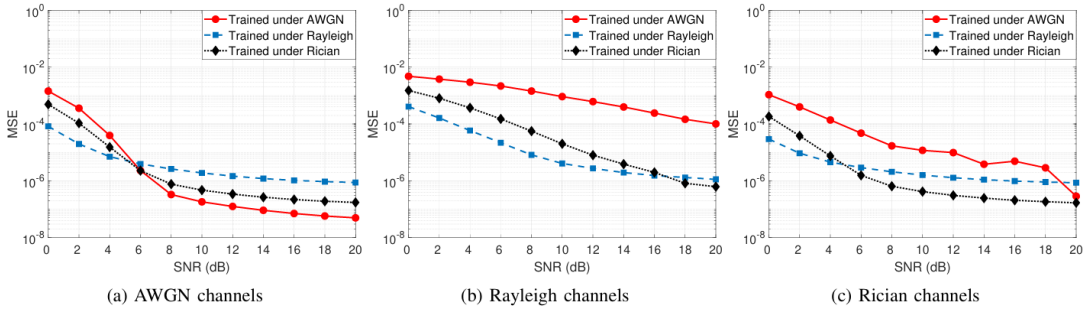


Figure 35: DeepSC-S MSE loss

For this reason, the Rician channel model is considered, and it can be observed that the MSE loss converges after about 400 epochs of training.

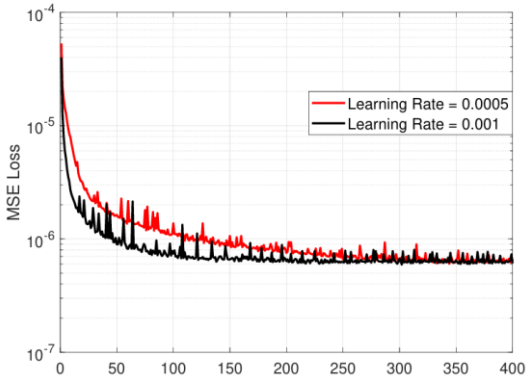
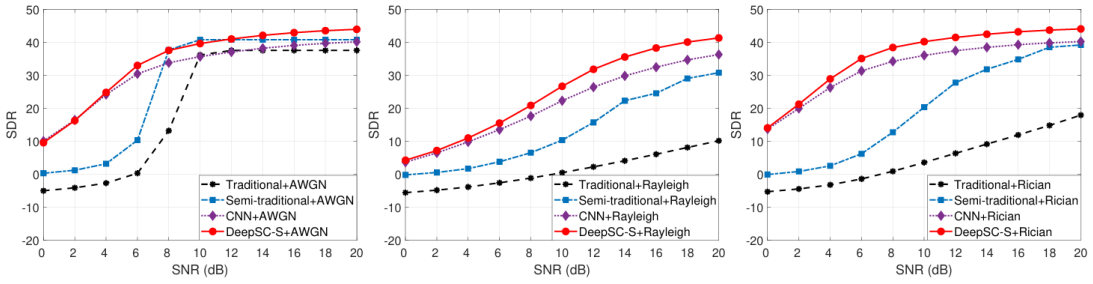


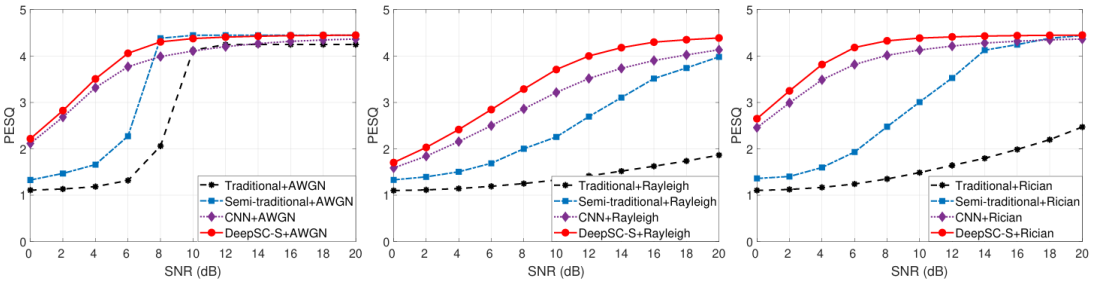
Figure 36: MSE Loss vs Epoch under the Richian channel with SNR = 8 dB

DeepSC-S is finally compared with a traditional communication system with extra feature coding for speech transmission, under AWGN, Rayleigh and Rician channels, and assuming accurate CSI knowledge.



(a) AWGN channels (b) Rayleigh channels (c) Rician channels

Figure 37: SDR score versus SNR for the different tested communication systems



(a) AWGN channels (b) Rayleigh channels (c) Rician channels

Figure 38: PESQ score versus SNR for the different tested communication systems

Plots in Figure **37** and Figure **38** show that DeepSC-S outperforms the traditional systems in terms of both SDR and PESQ scores. However, as for the previous case, no specific details are provided about the traditional and the semi-traditional systems employed; for this reason, further investigation are needed to draw definitive conclusions.

3. Methodology

The following sections provide the rationale behind the use case selection and a detailed description of the DL model used to test semantic transmission performance.

3.1 Use case: Text transmission

3.1.1 Rationale of Use Case

The use case selected for this thesis is text transmission in a 5G NR-based environment. Several considerations motivate this choice, which can be summarized as follows.

Text transmission inherently focuses on the accurate conveyance of meaning rather than on mere data exchange. Textual information consists of discrete symbols, i.e., words, each tied to specific semantic meanings. Thus, text can be considered an ideal candidate for exploring the nuances of semantic communication, since its goal is to ensure that the intended meaning is faithfully received and understood, rather than merely transmitting the source data.

Moreover, text is a fundamental form of communication, used across several domains, which highlights its relevance as a use case. By focusing on text, this research addresses an essential aspect of communication.

Additionally, text transmission, compared to audio and video transmission, is characterized by a relatively low level of complexity, making it an appropriate choice for an initial approach to semantic communication. Moreover, this reduced complexity facilitates the evaluation of key variables, making it easier to draw relevant conclusions.

Another key reason for choosing text-based communication is is its greater interpretability and explainability compared to other modalities such as audio or video.

The ability to analyse the reasons behind observed phenomena and to clearly depict the remarkable findings is fundamental in the context of a thesis project.

The listed factors collectively ensure that text transmission is a relevant and practical use case to understand the mechanisms that drive semantic communication.

The performed experiments aim to make a comparison between textual semantic transmission and a traditional 5G NR-based transmission chain, highlighting the eventual benefits of the new semantic paradigm.

3.2Adaptation of the DNN model

This section outlines the enhancements applied to the original DeepSC model. The goal is to assess the performance of the proposed semantic solution in an environment that more accurately simulates a modern communication system scenario. In fact, the original model presents three major limitations that prevent it from being directly compared with contemporary communication systems.

Problem 1: As described in the previous section, the original model implements three selectable physical channels as a network layer: the AWGN channel which adds noise to the transmitted symbols, the Rician channel and the Rayleigh channel, which simulate the effects of fading on the wireless transmission. To allow the model to better predict the behaviour of transmitted signals as they propagate through complex environments, a more accurate and realistic channel model is needed.

Problem 2: The model assumes a Single-Input Single-Output transmission system. While simpler, this assumption does not address the challenges posed by contemporary communication technologies, which predominantly utilize Multiple-Input Multiple-Output (MIMO) systems. MIMO has become a fundamental part of modern days communications, due to the higher data rate and reliability compared to SISO systems [92]. Moreover, since the scope of this study is to test the performance of a semantic communication system in a 5G New Radio-based scenario and to compare it to the traditional 5G New Radio transmission chain, the implementation of MIMO capabilities is fundamental for the scope of this project.

Problem 3: The simulator does not implement pre-coding or equalization techniques, which are crucial for mitigating inter-symbol interference, especially in a MIMO scenario. Consequently, these techniques are essential components of a realistic transmission system.

The remaining part of this section details the specific modifications made to transform the original model into a more robust and realistic version, enhancing its ability to provide more accurate simulation results.

3.2.1 Clustered Delay Line Channel Model

Problem 1 is tackled by introducing a Clustered Delay Line channel model (CDL). Clustered Delay Line models are defined by the European Telecommunication Standard Institute (ETSI) [93] and are particularly useful for modelling the multipath propagation environment.

In CDL models, the multipath components are organized into clusters, where each cluster represents a group of signal paths that arrive at the receiver; they mimic propagation mechanisms that characterize real life scenarios, such as reflections off the same or similar objects in the environment.

Each cluster is characterized by:

- a normalized delay, i.e. the delay of a certain cluster with respect to the earliest arriving cluster;
- a power in dB, which is the cluster signal power;
- an angle of departure (AOD), that is the azimuth angle at which a signal departs from the transmitter, measured in the horizontal plane from a reference direction;
- an angle of arrival (AOA), that is the azimuth angle at which a signal arrives at the receiver, measured in the horizontal plane from a reference direction;
- a zenith angle of departure (ZOD), the angle at which a signal departs from the transmitter, measured in the vertical plane from the zenith;

- a zenith angle of arrival (ZOA), the angle at which a signal arrives at the receiver, measured in the vertical plane from the zenith.

A total of five CDL models exist, which can be divided into two groups: Non-Line-Of-Sight (NLOS) profiles, that are CDL-A, CDL-B, CDL-C, and Line-Of-Sight (LOS) profiles, that are CDL-D and CDL-E. The first group simulate an urban environment, with more obstacles and, therefore, a more challenging scenario, while the second group is more suitable to represent high quality channels with a more direct signal path.

For this thesis, a CDL-B model is used, since it strikes a balance in complexity between CDL-A and CDL-C. The two LOS models are excluded since a clear and high-quality path does not accurately represent urban scenarios, which is the primary context for the considered use case.

Cluster #	Normalized delay	Power in [dB]	AOD in [°]	AOA in [°]	ZOD in [°]	ZOA in [°]
1	0.0000	0	9.3	-173.3	105.8	78.9
2	0.1072	-2.2	9.3	-173.3	105.8	78.9
3	0.2155	-4	9.3	-173.3	105.8	78.9
4	0.2095	-3.2	-34.1	125.5	115.3	63.3
5	0.2870	-9.8	-65.4	-88.0	119.3	59.9
6	0.2986	-1.2	-11.4	155.1	103.2	67.5
7	0.3752	-3.4	-11.4	155.1	103.2	67.5
8	0.5055	-5.2	-11.4	155.1	103.2	67.5
9	0.3681	-7.6	-67.2	-89.8	118.2	82.6
10	0.3697	-3	52.5	132.1	102.0	66.3
11	0.5700	-8.9	-72	-83.6	100.4	61.6
12	0.5283	-9	74.3	95.3	98.3	58.0
13	1.1021	-4.8	-52.2	103.7	103.4	78.2
14	1.2756	-5.7	-50.5	-87.8	102.5	82.0
15	1.5474	-7.5	61.4	-92.5	101.4	62.4
16	1.7842	-1.9	30.6	-139.1	103.0	78.0
17	2.0169	-7.6	-72.5	-90.6	100.0	60.9
18	2.8294	-12.2	-90.6	58.6	115.2	82.9
19	3.0219	-9.8	-77.6	-79.0	100.5	60.8
20	3.6187	-11.4	-82.6	65.8	119.6	57.3
21	4.1067	-14.9	-103.6	52.7	118.7	59.9
22	4.2790	-9.2	75.6	88.7	117.8	60.1
23	4.7834	-11.3	-77.6	-60.4	115.7	62.3

Per-Cluster Parameters					
Parameter	c_{ASD} in [°]	c_{ASA} in [°]	c_{ZSD} in [°]	c_{ZSA} in [°]	XPR in [dB]
Value	10	22	3	7	8

Figure 39: CDL-B cluster parameters

A 3GPP compliant CDL-B simulator built in MATLAB is used for the scope. This simulator can generate channel matrices containing the CDL-B coefficients with dimensionality [4 x 32], where 4 is the number of antennas equipped to each user and 32 is the number of antennas at the transmit side. For the purpose of this thesis, a total of 40000 channel realization are generated, i.e. 10 simulations of 4000 time slots (2 seconds, as detailed in Table 2), which correspond in total to a simulation of 20 seconds. The scenario considered is limited to a downlink transmission, with a base station (BS) at the transmit side and a mobile use (UE) at the receiver side.

This implementation provides DeepSC model with a more realistic physical channel layer.

3.2.2 Multiple-Input Multiple-Output (MIMO) Transmission

Problem 2 is solved by properly implementing MIMO transmission in DeepSC model.

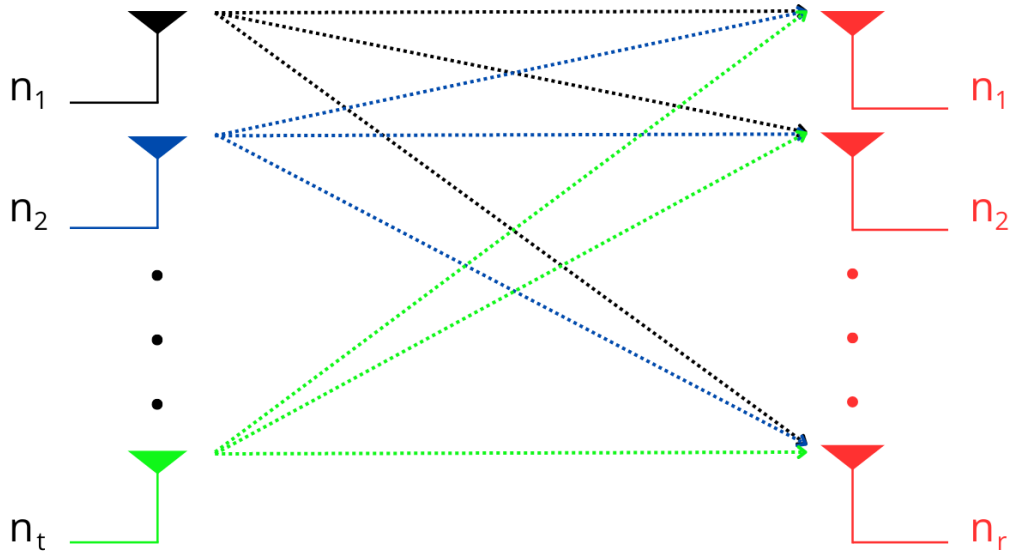


Figure 40. MIMO system representation, on the left side the transmitting antennas, on the right side the receiving antennas

MIMO technology, which relies on the usage of several transmitting antennas n_t and receiving antennas n_r operating at the same time on the same frequencies, allows to increase the spectral efficiency of the system. This results in significant improvements in data throughput, coverage, and reliability.

MIMO can operate in two different modes, which are Spatial multiplexing (SM) and Space Frequency Coding (SFC)

SM allows to simultaneously transmit multiple independent data flows on different spatial paths. In this operating mode, each antenna sends a different data stream; at the receiver, each stream is captured by multiple antennas and advanced space-time algorithms are needed to separate the different data flows. The maximum number of separable streams is given by the minimum between the number of receiving antennas and the number of transmitting antennas. This approach dramatically increases the system capacity, since the total data rate is multiplied by the number of data flows.

The system capacity, which in SISO systems can be described as

$$C \approx W \cdot log_2(1 + SNR) \quad [bit/s] \tag{64}$$

Adopting MIMO SM mode, the system capacity becomes

$$C \approx W \cdot N \cdot log_2(1 + SNR) \quad [bit/s] \tag{65}$$

where N is given by $min(n_t, n_r)$.

SFC mode, instead, enhances the reliability and the robustness of the transmission for a single data flow. Different copies of the same data flow are transmitted through different antennas, i.e., different spatial paths; therefore, the receiving antennas recover all the same data. This operation ensures that, even if the transmitted signal is degraded during their propagation in the wireless channel over a certain path, there is probably another path, or more, where the signal quality makes up for this loss. In this way, the overall information can still be recovered by the appropriate algorithms at the receiver.

SM and SFC serve different purposes, both of which are fundamental in modern communication systems. However, for the purpose of this thesis, the SFC mode is implemented in the semantic model, since the primary focus is to evaluate the reliability of the semantic transmission paradigm.

Considering the modified DeepSC implementation with MIMO support, each batch of encoded sentences $\boldsymbol{X} \in \mathcal{R}^{BxNLx2}$, is reshaped as $\boldsymbol{X} \in \mathcal{R}^{BNLx2}$: this flattening on two dimensions, allows the handling of multiple copies of the same batch.

Signal $\boldsymbol{X}$ is then properly pre-coded (more details are provided in the next sub-section), obtaining a new pre-coded signal $\boldsymbol{X_{tx}} \in \mathcal{R}^{n_t xBNLx2}$, that is transmitted over the physical channel. The recovered message at the receiver side, can be denoted as

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X_{tx}} + \boldsymbol{n} \tag{66}$$

where $\boldsymbol{n}$ is the introduced noise vector. Depending on desired configuration, the channel matrix $\boldsymbol{H}$ can include the CDL-B coefficients computed by the aforementioned MATLAB simulator or can collect random variables that statistically represent Rician or Rayleigh fading channels. In particular, the Rician channel matrix is composed of random variables with mean $\sqrt{\frac{K}{K+1}}$ and standard deviation $\sqrt{\frac{1}{K+1}}$, where K is the Rician K factor, defined as:

$$\boldsymbol{K} = \frac{\boldsymbol{Power\ of\ LOS\ components}}{\boldsymbol{Power\ of\ NLOS\ components}} \tag{67}$$

The Rayleigh coefficients are, instead, realizations of a random variable with mean 0 and standard deviation $\frac{1}{2}$. The AWGN channel, instead, does not implement MIMO transmission.

The received signal is then demodulated by implementing equalization techniques, which are detailed in the next sub-section.

With the enhancements described, DeepSC now support both SISO and MIMO transmissions. In particular, the simulator supports a maximum of 32 n_t and 4 n_r.

3.2.3 Pre-coding and Equalization Techniques

Lastly, pre-coding and equalization techniques have been introduced to increase the system robustness.

The implemented pre-coding techniques assumes that the channel matrix $\boldsymbol{H}$ is perfectly known at the transmitter. By means of the single values decomposition (SVD), it is possible to get

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathbf{H}} \tag{68}$$

where, given that $\boldsymbol{H}$ is a $\boldsymbol{m\,x\,n}$ matrix, $\boldsymbol{U}$ is an orthogonal matrix with dimensions $m\,x\,m$, where the columns are the left singular vectors of $\boldsymbol{H}$, $\boldsymbol{\Sigma}$ is an $\boldsymbol{m\,x\,n}$ matrix containing the singular values of $\boldsymbol{H}$ in descending order and $\boldsymbol{V^H}$ is an $n\,x\,n$ matrix containing the right singular vectors.

The matrix $\boldsymbol{V}$, obtained as $\boldsymbol{V} = (\boldsymbol{V^H})^{\boldsymbol{H}}$ is the pre-coding matrix used at the transmitter. The sequence of symbols to be transmitted is multiplied by the pre-coding matrix before the transmission. Hence, the transmitted signal $\boldsymbol{X_{tx}}$ can be written as:

$$\boldsymbol{X_{tx}} = \mathbf{V}\mathbf{X}_{\mathbf{MIMO}} \tag{69}$$

Algorithm 6: Pre-coding in a MIMO setting
Input: The sample batch $\boldsymbol{B}$, the channel matrix $\boldsymbol{H}$, the batch of encoded sentences $\boldsymbol{X}$
Output: The pre-coded signal $\boldsymbol{X_{tx}}$
1: Perform $SVD(\boldsymbol{H}) \rightarrow \mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}^{\mathbf{H}}$ **2:** Compute $\boldsymbol{V} = (\boldsymbol{V^H})^{\boldsymbol{H}}$ **3:** Re-shape $\boldsymbol{X} \rightarrow \boldsymbol{X} \in \mathcal{R}^{BNLx2}$ **4:** Generate n_t copies of $\boldsymbol{X} \rightarrow \boldsymbol{X_{MIMO}} \in \mathcal{R}^{n_t xBNLx2}$ **5:** Generate the pre-coded signal as $\boldsymbol{X_{tx}} = \boldsymbol{V}\boldsymbol{X_{MIMO}}$

To mitigate the channel distortions, two different equalization techniques are implemented, depending on the considered type of channel.

Zero-Forcing (ZF) Equalization:

ZF equalization is a linear equalization technique used to mitigate the Inter-Symbol Interference (ISI), which arise when multiple signals are transmitted simultaneously over the same channel, interfering with each other.

In the context of MIMO systems, the ZF equalization is carried out by multiplying the inverse of the channel matrix by the received signals, effectively compensating for the interference components affecting each antenna.

In the context of this project, since the transmitted signal is pre-coded, the received signal $\boldsymbol{Y}$ can be re-written as

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X}_{\boldsymbol{tx}} + \boldsymbol{n} = \boldsymbol{H}\boldsymbol{V}\boldsymbol{X}_{\boldsymbol{MIMO}} + \boldsymbol{n} \tag{70}$$

Therefore, recalling (61), the ZF equalization can be carried out by multiplying the received signal by $\boldsymbol{U}^{\boldsymbol{H}}$ and $\boldsymbol{V}^{\boldsymbol{H}}$, obtaining

$$\boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{\Sigma}^{\boldsymbol{H}}\boldsymbol{Y} = \boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{\Sigma}^{\boldsymbol{H}}\boldsymbol{H}\boldsymbol{V}\boldsymbol{X}_{\boldsymbol{MIMO}} + \boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{\Sigma}^{\boldsymbol{H}}\boldsymbol{n} =$$
$$\boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{\Sigma}^{\boldsymbol{H}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{X}_{\boldsymbol{MIMO}} + \boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{\Sigma}^{\boldsymbol{H}}\boldsymbol{n} = \boldsymbol{X}_{\boldsymbol{MIMO}} + \boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{\Sigma}^{\boldsymbol{H}}\boldsymbol{n} \tag{71}$$

ZF equalization nullifies the effects of ISI on the transmitted signal, however, as it can be observed in (64), it also amplifies the noise component of the received signal.

Minimum Mean Squared Error (MMSE) Equalization:

MMSE is a linear equalization technique with the goal of finding an equalization matrix $\boldsymbol{W}_{\boldsymbol{MMSE}}$ that minimizes the mean squared error between the transmitted signal, in the context of this thesis the encoded sample batch $\boldsymbol{X}$ is considered, and the estimated signal at the receiver expressed as $\widehat{\boldsymbol{X}} = \boldsymbol{W}_{\boldsymbol{MMSE}} \cdot \boldsymbol{Y}$, where $\boldsymbol{W}_{\boldsymbol{MMSE}}$ corresponds to

$$\boldsymbol{W}_{\boldsymbol{MMSE}} = \boldsymbol{argmin} \left|\left| \boldsymbol{X} - \widehat{\boldsymbol{X}} \right|\right|^{\boldsymbol{2}} \tag{72}$$

By minimizing the mean squared error between the transmitted and the received signal, the MMSE equalizer strikes a balance between eliminating ISI and controlling noise amplification. This is in contrast with the Zero-Forcing (ZF) equalizer, that only focuses on inverting the channel matrix to eliminate ISI, being, therefore, less robust against noise.

The MMSE equalizer matrix can be computed as follows:

$$\boldsymbol{W_{MMSE}} = \boldsymbol{H_{eq}^H} * \left(\boldsymbol{H_{eq}^H} * \boldsymbol{H_{eq}} + (\boldsymbol{N_0} * \boldsymbol{I})\right)^{-1} \tag{73}$$

where $\boldsymbol{H_{eq}}$ is the equivalent matrix computed as $\boldsymbol{H_{eq}} = \boldsymbol{VH}$ with shape $n_r \times 1$, N_0 is the noise power computed as the ratio of the received signal power and the SNR in linear units $N_0 = P_{rx}/SNR_{linear}$ and $\boldsymbol{I}$ is the identity matrix with shape $m \times m$, where m is the number of columns of $\boldsymbol{H_{eq}}$.

This type of equalization is more suitable to equalize the received signal when CDL-B channel model is considered, since a more robust protection against noise and ISI is required for a richer radio channel.

Algorithm 7: MMSE equalization
Input: The matrix **V**, the channel matrix $\boldsymbol{H}$, ***the noise power*** N_0 **Output:** The estimated signal $\widehat{\boldsymbol{X}}$ **1:** Compute the equivalent matrix $\boldsymbol{H_{eq}} = \boldsymbol{VH}$ **2:** Compute the hermitian of the equivalent matrix $\boldsymbol{H_{eq}^H}$ **3:** Generate the identity matrix I **4:** Compute the MMSE matrix as $\boldsymbol{W_{MMSE}} = \boldsymbol{H_{eq}^H} * \left(\boldsymbol{H_{eq}^H} * \boldsymbol{H_{eq}} + (\boldsymbol{N_0} * \boldsymbol{I})\right)^{-1}$ **5:** Compute the estimated signal as $\widehat{\boldsymbol{X}} = \boldsymbol{W_{MMSE}} \cdot \boldsymbol{Y}$

In conclusion, the implemented techniques simulate a more realistic MIMO transmission line. With all refinements previously described, the model is ready to be tested and compared with a more traditional communication system, i.e. a 5G New Radio-based simulator, to assess the performance of the updated model.

3.3 5G New Radio Simulator

To compare the semantic system to the contemporary telecommunication systems, a 5G NR simulator, courtesy of TIM, is used. In this section, a high-level description the simulator is provided.

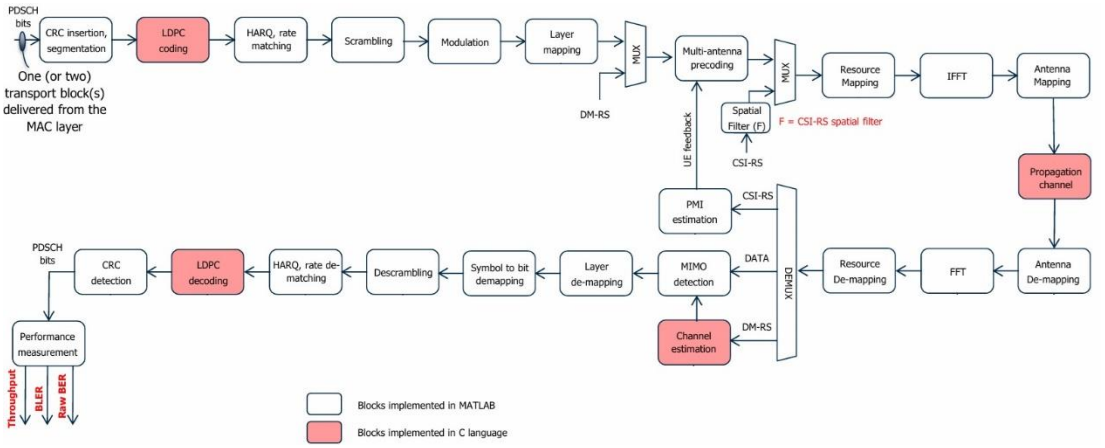


Figure 41. 5G NR simulator, PDSCH transmission chain

Figure 41 depicts the simulator building blocks in detail, as it can be seen, a mixture of MATLAB and C languages is used for the implementation. In particular, the transmission chain implements a Low-Density Parity-Check (LDPC) encoder, the rate-matching block used to adjust the coding rate, the Hybrid Automatic Repeat Request (H-ARQ) block responsible for the retransmission (up to 3 retransmissions per block) and the modulation block which supports several modulations (ranging from the QPSK to the 256-QAM). The combined action of the mentioned blocks makes possible for the simulator to implement the Adaptive Modulation and Coding Mechanism, which adjust the modulation and coding scheme (MCS) according to the channel conditions. Table 1 reports the available MCS according to the 5G 3GPP standard [94].

Table 1: MCS index table 2 for PDSCH

MCS Index I_{MCS}	Modulation Order Q_m	Target code Rate R x [1024]	Spectral efficiency
0	2	120	0.2344
1	2	193	0.3770
2	2	308	0.6016
3	2	449	0.8770
4	2	602	1.1758
5	4	378	1.4766
6	4	434	1.6953
7	4	490	1.9141
8	4	553	2.1602
9	4	616	2.4063
10	4	658	2.5703
11	6	466	2.7305
12	6	517	3.0293
13	6	567	3.3223
14	6	616	3.6094
15	6	666	3.9023
16	6	719	4.2129
17	6	772	4.5234
18	6	822	4.8164
19	6	873	5.1152
20	8	682.5	5.3320
21	8	711	5.5547
22	8	754	5.8906
23	8	797	6.2266
24	8	841	6.5703
25	8	885	6.9141
26	8	916.5	7.1602
27	8	948	7.4063
28	2	reserved	
29	4	reserved	
30	6	reserved	
31	8	reserved	

This model provides a 3GPP specification compliant radio interface and a reliable tool to evaluate 5G NR-based point-to-point communications between a NR base station (gNodeB) and a single User Equipment (UE) node, in SISO or MIMO mode.

The simulation framework makes use of two types of channel models, compliant to the 3GPP specifications, i.e. TDL and CDL. As previously discussed, the CDL-B model is used to perform the experiments. Additionally, the UE speed is assumed to be 5 Km/h, simulating a walking pedestrian.

Moreover, a Huffman encoder [95], and decoder, are used to encode the sentences in binary code, these blocks are also implemented in python. The encoder builds a

codebook containing the variable length binary encoding of each character that can be found in the test set. More specifically, more common symbols, characters in this case, are associated with fewer bits than less common symbols, generating a prefix-free binary tree, used then at the receiver side to decode the received bits and return the original characters.

Two additional blocks have been inserted in the transmission chain, the first one at the transmitter side organizes the bits in transport blocks (TB), the second one at the receiver side rearranges the TBs to manage the retransmissions. The final structure of the transmission chain is represented in Figure **42**.

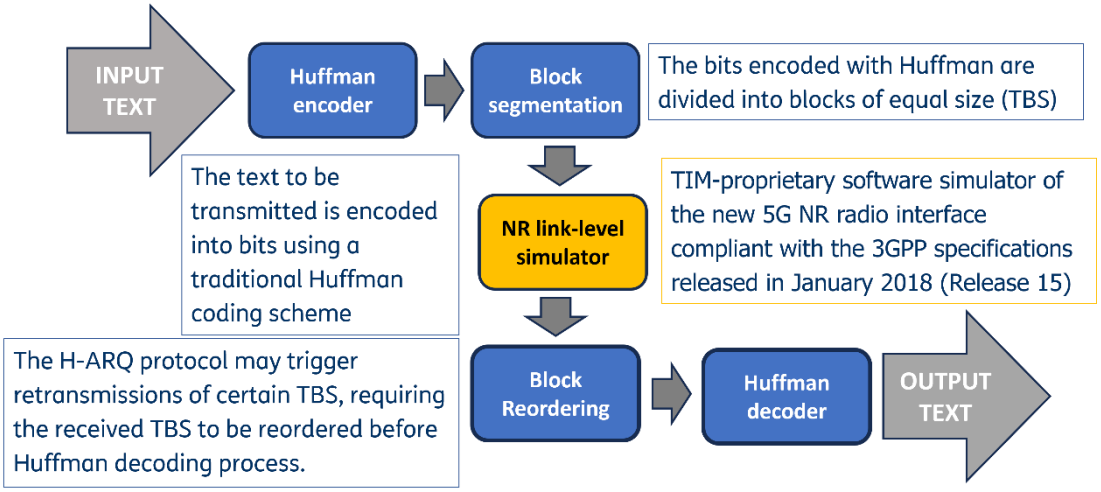


Figure 42: Representation of the modified simulator transmission line

Table 2 shows some of the parameters used during the simulations.

Table 2: Simulator Parameters

Parameter name	**Value**
System Bandwidth [MHz]	80
Sub-carrier spacing [kHz]	30
Slot duration [ms]	0.5
N. of TX antennas in vertical dimension per panel	4
N. of TX antennas in vertical dimension per panel	4
N. of antenna panels	1
N. of TX antennas	32

N. of RX antennas	4
N. of antennas elements in azimuth	4
N. of antennas elements in elevation	4
N. of polarizations	2
N. of codewords	1
N. of layers for first codeword	2
N. of resource blocks allocated	217
Max. H-ARQ transmission	4
MCS Table	256-QAM [Table 1]
Beamforming Scheme	'PMI-based'
Channel Model	'CDL_3GPP'
Channel Power Delay Profile	'CDL_B'
Mobile Speed [km/h]	5
Carrier Frequency [MHz]	3.64e3
Simulation length in number of slots	4000

4. Results

In this chapter the results achieved by the authors of [13] are presented and compared with those based on the improvement presented in Chapter 3.

Moreover, the modified DeepSC performance is compared with the 5G NR-based simulator over the text transmission task.

4.1 Results from the original model

In the reference paper [13], Xie et al. compare DeepSC other DNN algorithms and the traditional channel coding approaches under AWGN and Rayleigh fading channel, assuming perfect knowledge of Channel State Information (CSI) for all schemes.

Simulation Settings:

The adopted dataset is part of the proceedings of the European Parliament, consisting of several thousand of sentences, being pre-processed into sentences of lengths between 4 and 30 words, and more than 22 thousand different words.

As Table 3 shows, the DeepSC model includes three transformer encoder layers and three transformer decoder layers, which are set with 8 heads for the multi-head attention mechanism and 128 units, with a linear activation function. At the transmitter side, the two dense layers are set with 256 units and 16 units respectively, while at the receiver side the first dense layer is set with 256 units and the second one with 128 units. The MI model makes use of two dense layer set at 256 units to extract the information and one dense layer with 1 unit to integrate the information.

Table 3: DeepSC model settings

	Layer Name	Units	Activation
Transmitter	3×Transformer Encoder	128 (8 heads)	Linear
	Dense	256	ReLu
	Dense	16	ReLu
Channel	AWGN/Rician/Rayleigh	None	None
Receiver	Dense	256	ReLu
	Dense	128	ReLu
	3×Transformer Encoder	128 (8 heads)	Linear
	Prediction Layer	Dictionary Size	Softmax
MI Model	Dense	256	ReLu
	Dense	256	ReLu
MI Model	Dense	1	ReLu

Both Joint source-channel coding based on neural network and typical methods to separate source and channel coding are analysed:

- DNN based JSSC [96] where the network consists of Bi-directional Long Short-Term Memory (BLSTM) layers, labelled as JSCC in the following figures
- The traditional methods where source and channel coding are separated use the following technologies:
  - Source coding: Huffman coding, Brotli coding and fixed-length (5-bit) coding
  - Channel coding: Turbo coding and Reed-Solomon (RS) coding

The metrics used to evaluate the system performance are the BLEU score and the sentence similarity score.

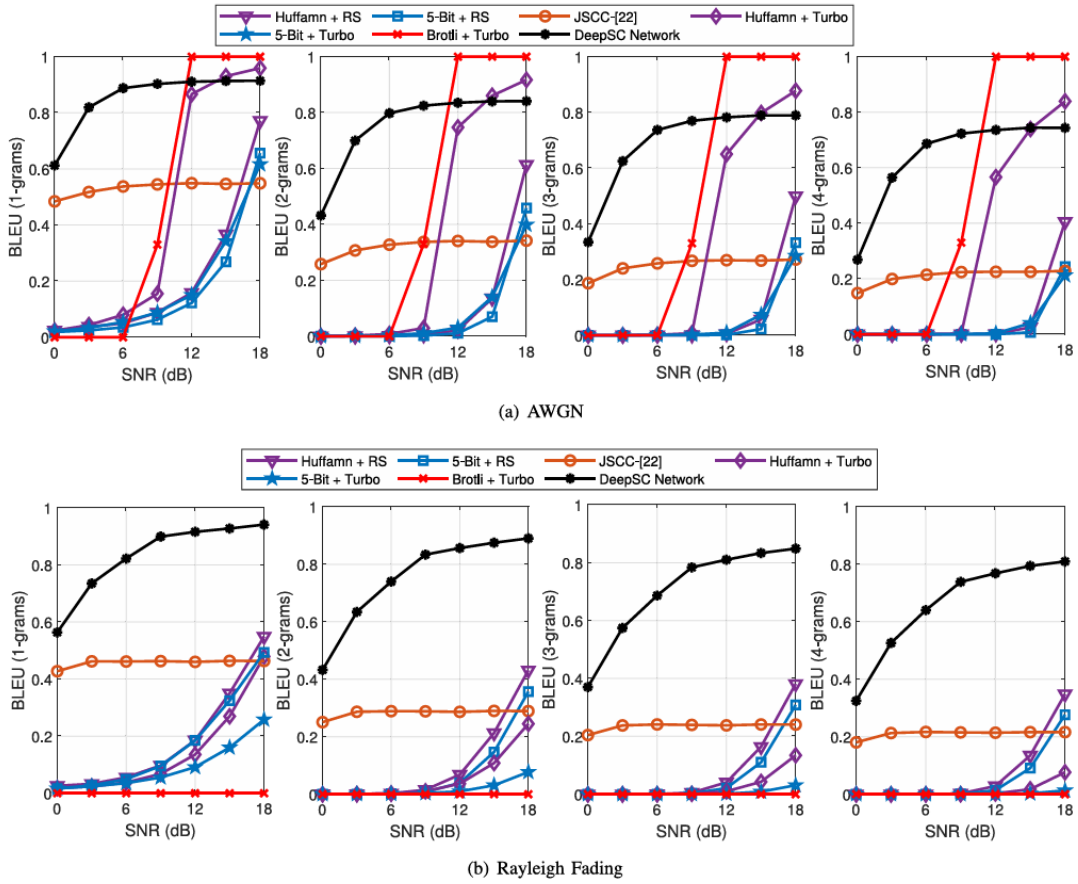


Figure 43. BLEU score versus SNR [13]

Figure *2*Figure **43** displays the BLEU score performance, considering different n-grams values, of the considered systems, under the same number of transmitted symbols, over a range of SNR values. More specifically, the proposed DeepSC is compared to the DNN base JSCC, both trained over the AWGN and Rayleigh fading channels, to Huffman coding + RS coding in 64-QAM, 5-bit coding with RS coding in 64-QAM, Huffman coding with Turbo coding in 64-QAM, 5-bit coding with Turbo coding in 128-QAM and Brotli coding with Turbo coding in 8-QAM.

On the AWGN channels, Brotli + Turbo and Huffman + Turbo outperform the other approaches when the SNR is higher than 12 dB, due to the decreased channel distortion. However, in the low SNR regime, DL enabled approaches perform better.

On the Rayleigh fading channel, instead, DeepSC outperforms all the other approaches regardless of the SNR value or the number of n-grams considered.

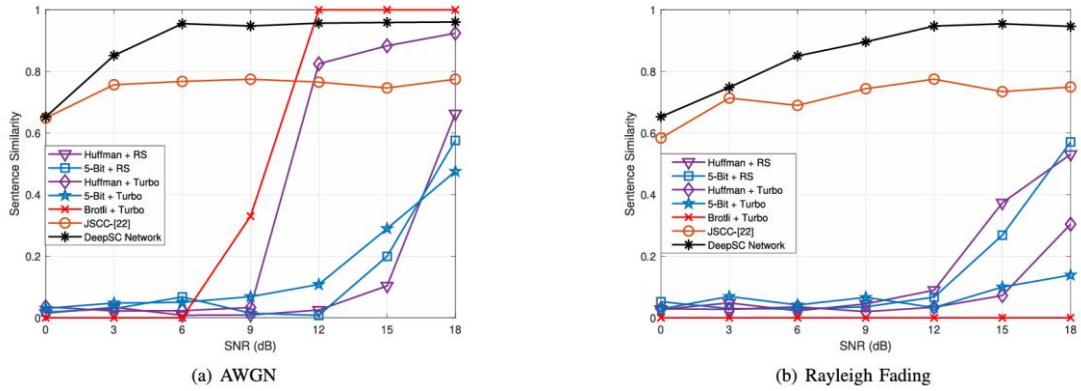


Figure 44. Sentence similarity versus SNR [13]

Figure **44** displays the sentence similarity score of the considered model, under the same number of symbols, in a certain SNR range and over the AWGN and the Rayleigh Fading channels. Figure **44** (a) and (b) show similar tendencies to Figure **43** (a) and (b). In Table 4 few representative results, obtained over Rayleigh fading channel and SNR 18 dB, are shown.

Table 4: Example of a reconstructed sentence with different methods

Transmitted sentence	it is an important step towards equal rights for all passengers.
DeepSC	it is an important step towards equal rights for all passengers.
JSCC	it is an essential way towards our principles for democracy.
Huffman + Turbo	rt is an imeomant step tomdrt equal rights for atp passurerrs.
Huffman + RS	it is an important step towards ewiral rlrsuo for all passengess.
Bit5 + Turbo	it is an yoportbnt ssep sowart euual qighd fkr ill passeneers.
Bit5 + RS	It iw an ymp!rdbnd stgo to!atds eq.al ryghts dkr alk passengers.

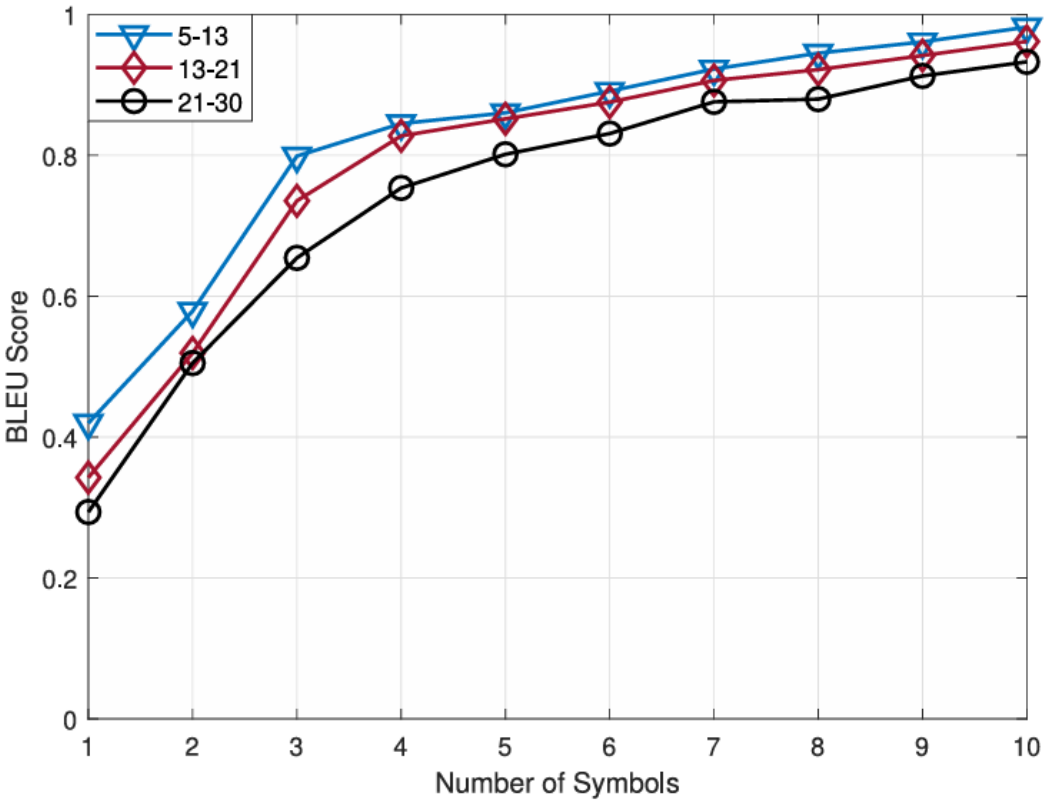


Figure 45. BLEU score (1-gram) versus the average number of symbols used to represent a word with SNR = 12 dB [13]

Figure 45 displays how a greater number of symbols is beneficial for DeepSC, in fact, as the number of symbols used to represent a word increases the distance between constellations increases too. Moreover, this figure highlights the DeepSC difficulties to understand longer sentences, since the sentence structure becomes more complex.

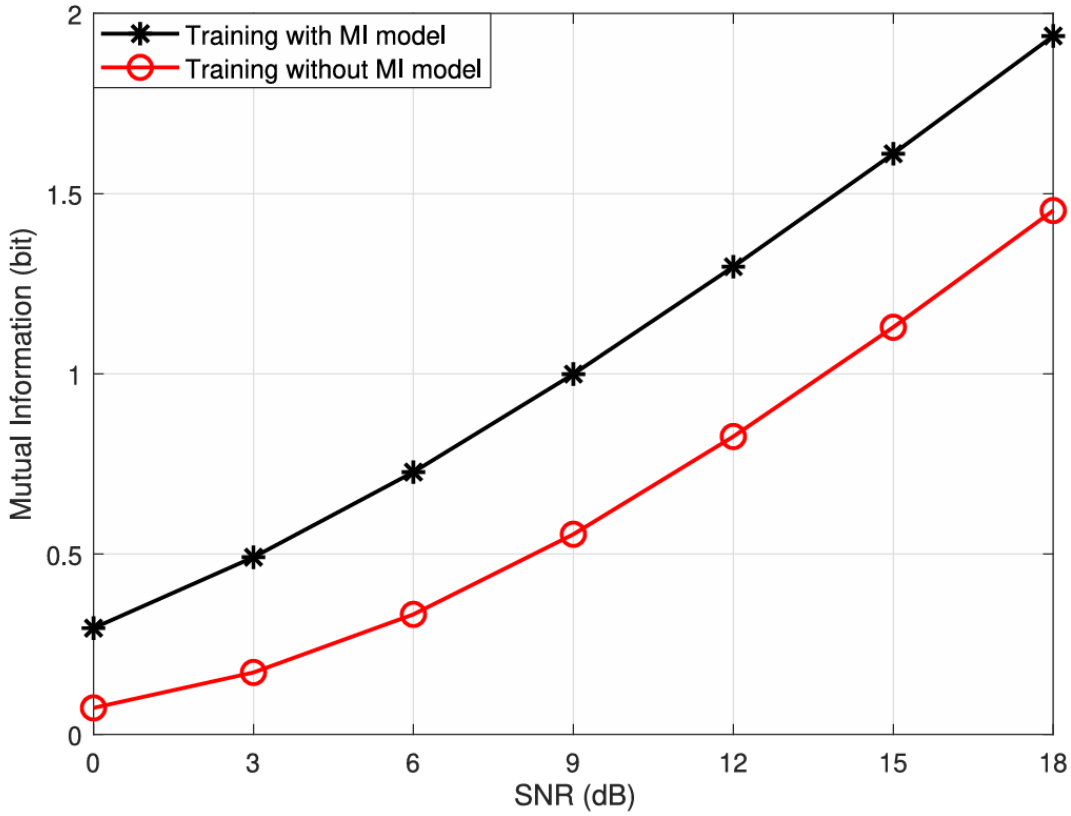


Figure 46. SNR vs MI for different trained encoders [13]

Figure **46**Figure **47**demonstrates the relationship between the mutual information and SNR after training. It appears clear that the encoder trained with the MI model outperforms the one trained without it, demonstrating the benefits of incorporating the mutual information into the system's loss function.

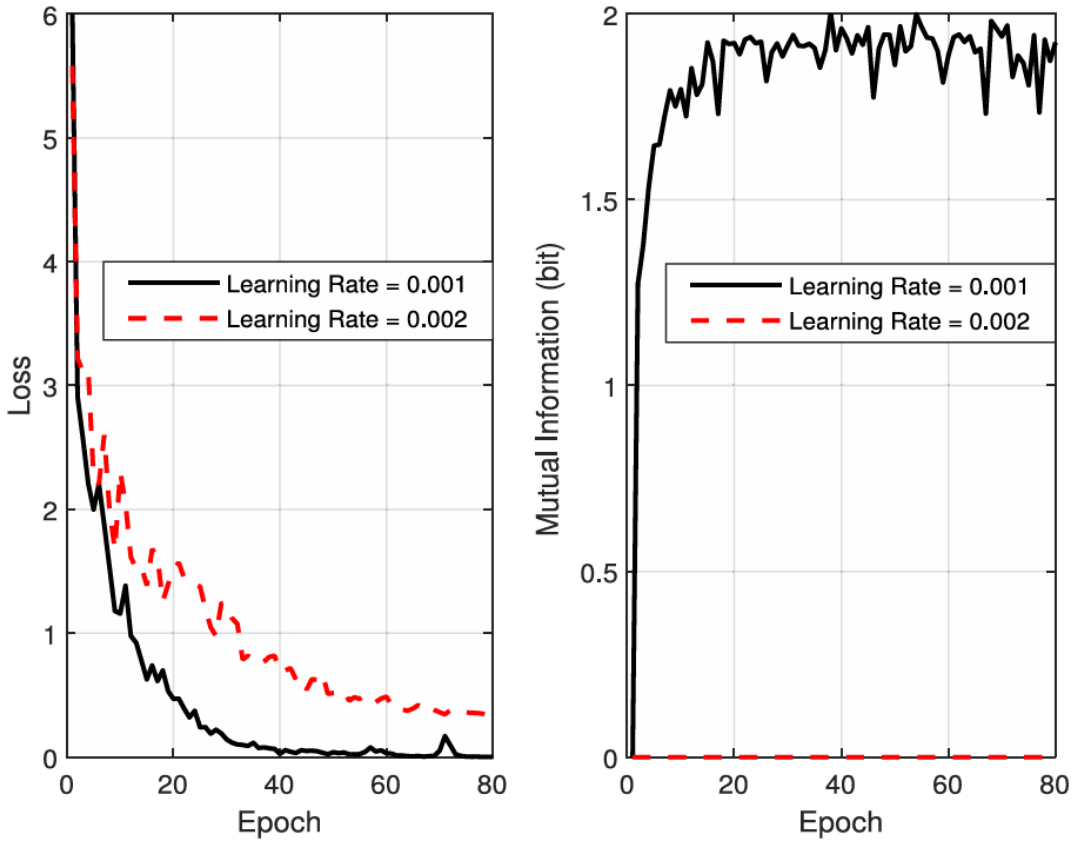


Figure 47. Impact of different learning rates with training, SNR = 12 dB [13]

Figure 47 is useful to understand the relationship between the loss value and the mutual information during the training: it can be seen as after epoch 40 the loss and the mutual information float around the same values.

Finally, the authors provide a complexity analysis in terms of the average processing runtime per sentence, excluding the runtime of source coding and decoding.

	DeepSC	JSCC	RS coding	Turbo coding
Runtime	3.27 ms	2.71 ms	4.14 ms	8.59 ms

From the table, the DL enabled approaches have lower runtime than the traditional approaches, with JSCC requiring the lowest average runtime due to its simple

architecture, at the expense of semantic processing capabilities. DeepSC on the other hand, performs better than the traditional approaches.

Considering the achieved results, DeepSC seems a promising semantic model. However, the settings of the traditional approaches are not stated explicitly in the reference paper [13]. It should be noted that results obtained with traditional approaches are suspiciously underperforming, e.g., the Turbo+Huffman scheme shows poor performances even with rather high Signal to Noise ratio, which suggests suboptimal radio chain configuration. For instance, it is not explained how, and if, the retransmission of the traditional approaches is managed, and the code-rate is not mentioned for any of them. Therefore, a deeper knowledge of the testing conditions would be needed to draw meaningful conclusions.

4.2 Results from the updated model

In this section the results obtained using the DeepSC model, original model and updated model, and the 5G NR simulator are presented.

The following DeepSC results are obtained using both SISO and MIMO (32×4) systems, over the Rician fading channel, the Rayleigh fading channel and the CDL-B channel models. For the DeepSC case, the dataset of CDL-B matrices, obtained through the TIM-proprietary simulator, is split into training, validation and test sets, since it is a good practice to validate and test the model using different data with respect to the data used during the training phase.

During the training phase, the SNR changes in a range from 5 dB to 10 dB: this is done to enable the models to perform effectively in both low and high SNR regimes. In the validation phase, the SNR is fixed to 20 dB to evaluate the model evolution while it is not affected by the channel noise, while the test process considers SNR values ranging from -4 dB to 20 dB.

The transmitted signal and the channel matrix coefficients are normalized to ensure that the signal power at the transmitter and the average power of the channel coefficients is equal to 1.

The first test is carried out on the SISO systems. In Figure 48, the two graphs represent the 1-gram BLEU score achieved on the considered channels: as it can be seen, the Rayleigh channel and the Rician channel perform slightly better than the CDL-B one. In fact, with respect to the other mentioned channels, the CDL-B channel is richer, i.e. the multi-path components are more prominent, therefore, its performance is expected to be worse. Moreover, the BLEU score is greatly affected by the SNR value. Note that, the obtained results are compatible with the ones from the reference paper [78]. However, since during this experiment an MMSE equalization is assumed, instead of a ZF equalization, the obtained results are slightly better.

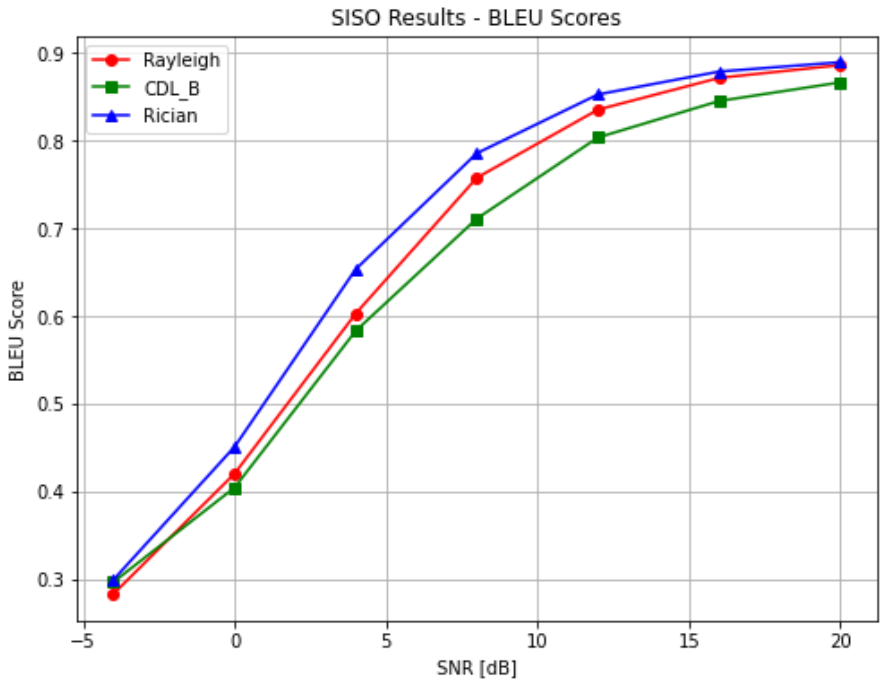


Figure 48: SISO BLEU score (4 n-gram).

Considering the Sentence similarity score (Figure 49) the results on the three channels are similar to the ones already seen in the BLEU score plot. The fact that the achieved values are similar means that the occurred errors do not have a big impact on the model ability to recognize semantic patterns in the received sentences. It is worth noting that, even at low SNR values, the sentence similarity score remains above 90% across all

channel conditions. This indicates that the meaning of the transmitted sentences is generally conveyed accurately, even in particularly challenging scenarios.

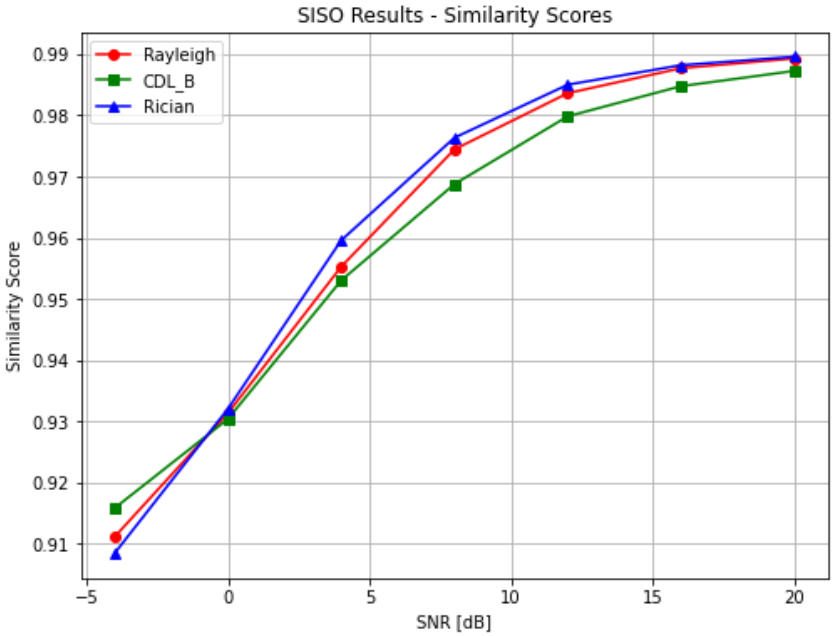


Figure 49: SISO sentence similarity score.

MIMO system results (Figure 50) show a significant difference in performance in a low SNR regime: while the model performs similarly across the three channels when SNR exceeds 5 dB, it shows better performance in CDL-B channel conditions compared to the Rayleigh channel at lower SNR values. These results can be explained by considering that MIMO systems gain greater advantages in environments with rich multipath conditions. Moreover, it can be observed how MIMO improves the model performance, since the achieved BLEU scores are much higher than the ones obtained in the SISO case.

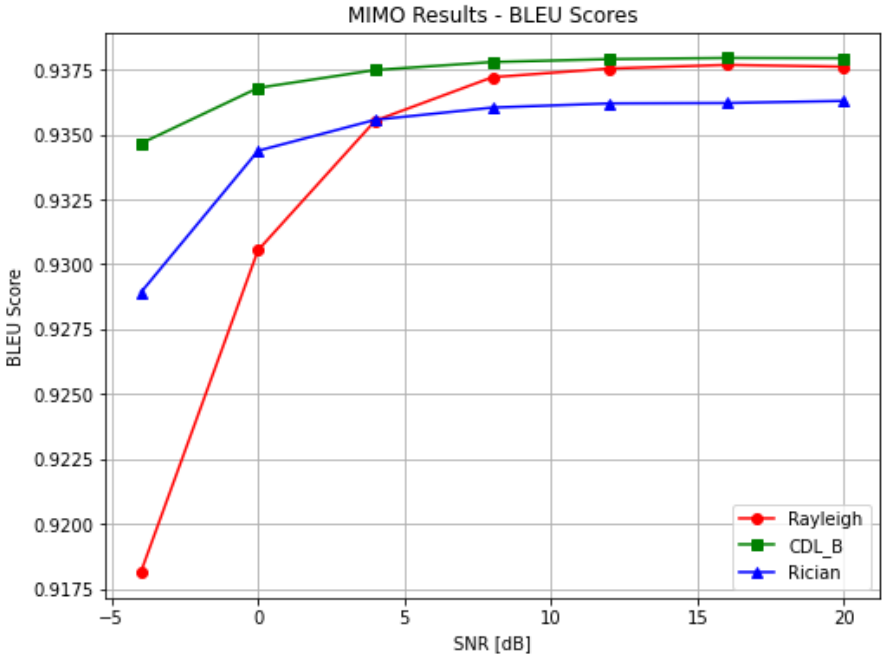


Figure 50: MIMO BLEU score (4 n-gram).

Similar conclusions can be drawn from the comparison of the sentence similarity scores (Figure **51**). For the CDL-B scenario, the model offers slightly better performance; however, it shows very good performance in terms of semantic reconstruction of the transmitted sequence across all considered channel models.

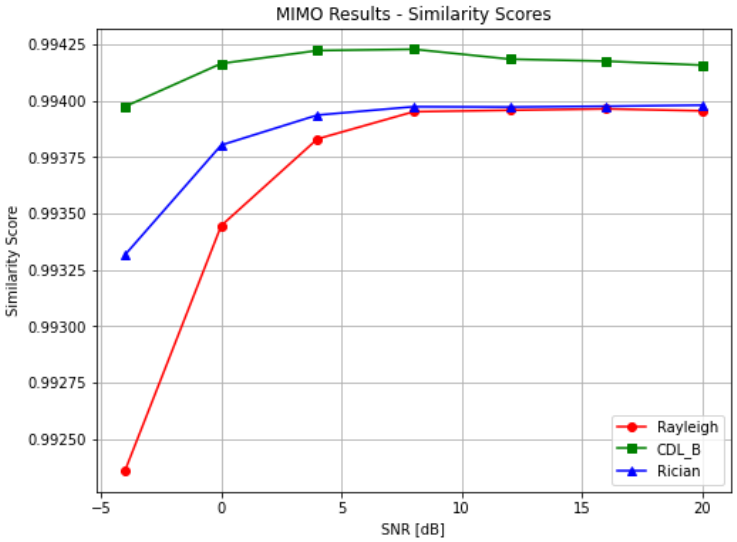


Figure 51: MIMO sentence similarity score.

Figure **52** displays the relationship between the mutual information and the SNR estimated at the receiver side. As expected, MI appears directly proportional to SNR, except for several outlier values observed mainly during the early epochs of training. As training progresses, the MI neural network provides increasingly reliable estimates. Moreover, the advantages of a MIMO system become evident, as it enhances mutual information and, consequently, boosts channel capacity.

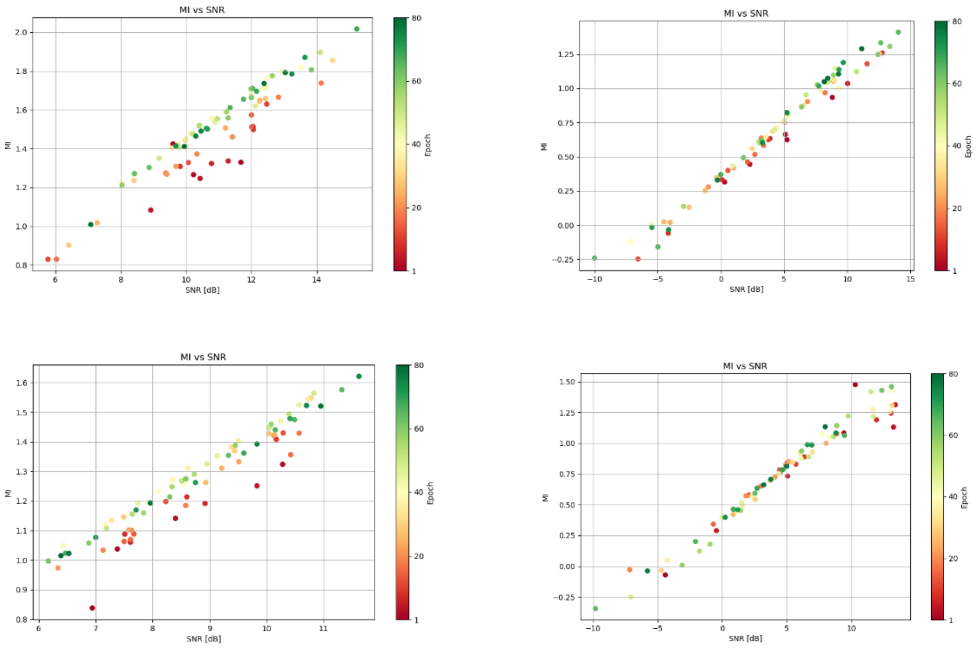


Figure 52: MI VS SNR graph of CDL-B MIMO (top-left), CDL-B SISO (top-right), Rayleigh MIMO (bottom-left), Rayleigh SISO (bottom-right).

The comparison between the original and the modified DeepSC models shows the benefits of the introduced changes. In the following part of this section, the new DeepSC is compared to the traditional 5G-based solution based on the NR link simulator described in 3.3. To ensure a fair comparison between the two models, both are tested using the CDL-B channel.

Errore. L'autoriferimento non è valido per un segnalibro. reports the BLER values for each retransmission performed by the simulator at different SNR values.

Table 5: 5G NR simulator BLER vs SNR

MCS	SNR [dB]	BLER -TX attempt 1	BLER - First Retransmission	BLER - Second Retransmission	BLER -Third Retransmission
3	-4	0.010897	0	0	0
5	0	0.120472	0	0	0
7	4	0.094076	0	0	0
8	8	0.045872	0	0	0
12	12	0.096756	0	0	0
14	16	0.109597	0	0	0
15	20	0.083379	0	0	0

Figure 53**Errore. L'origine riferimento non è stata trovata.** displays a comparison between the BLEU scores and the similarity scores achieved by the 5G NR-based transmission chain and the modified DeepSC model. It is noteworthy that the simulator outperforms the proposed semantic model, primarily due to its reliance on the H-ARQ retransmission mechanism, which effectively addresses transmission errors.This capability, along with other factors, contributes to the perfect reconstruction of the sentence at the receiver. The results show clearly that, for the considered use case, the proposed new DeepSC model is a sub-optimal solution compared to the traditional approach in terms of reliability.

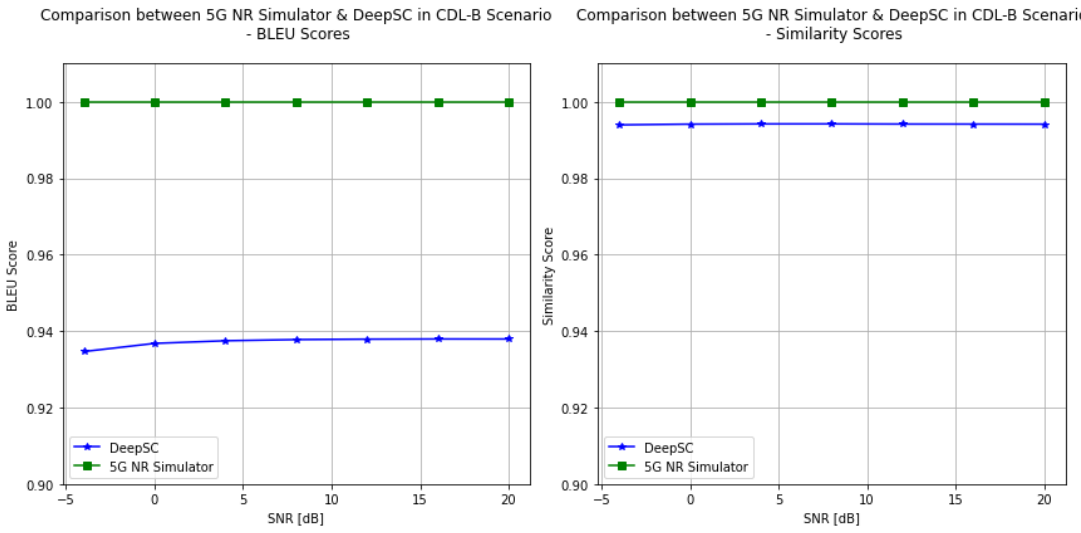


Figure 53: Comparison between the 5G NR Simulator & the DeepSC model

Figure 54 presents a comparison of the average resource elements allocated per transmitted word by the two approaches. DeepSC encodes each word using 8 complex symbols, regardless of the channel conditions. In contrast, the 5G NR simulator employs an adaptive coding and modulation technique that adjusts the number of resource elements allocated per word based on channel quality, specifically the SNR. When the SNR is higher, fewer parity bits are required, resulting in the encoding of words with fewer resource elements. Conversely, in a low SNR regime, where channel quality is poor, the simulator allocates more resource elements per word. To make these assessments, the average number of resource elements per word $\widehat{N_{RE}}$ is computed as follows:

$$\widehat{N_{RE}} = \left(N_{subcarriers} \times N_{OFDM\ Symbols} \times N_{Resource\ Blocks\ Allocated}\ x\ N_{Timeslots}\right) \div N_{words} \quad (67)$$

Table 2 reports the value assigned to each term in the formula. N_{words} is the total number of transmitted words and can be computed as

$$N_{words} = N_{Transmissions} \times N_{test\ set\ sentences} \times L_{sentence} \quad (68)$$

Where $N_{Transmissions}$ is the total number of transmissions occurred during the simulation time (it depends on the BLER, therefore this value changes with the SNR), $N_{test\ set\ sentences} = 4000$ is the number of sentences in the test set, $L_{sentence} = 30$ is the length in terms of words of each sentence of the test set.

The results clearly show that in high SNR regimes, the traditional method performs better than DeepSC. However, in low SNR regime, DeepSC uses less symbols per word on average, meaning that the proposed model produces good results, in terms of sentence reconstruction quality, while utilizing less bandwidth than the traditional

method.

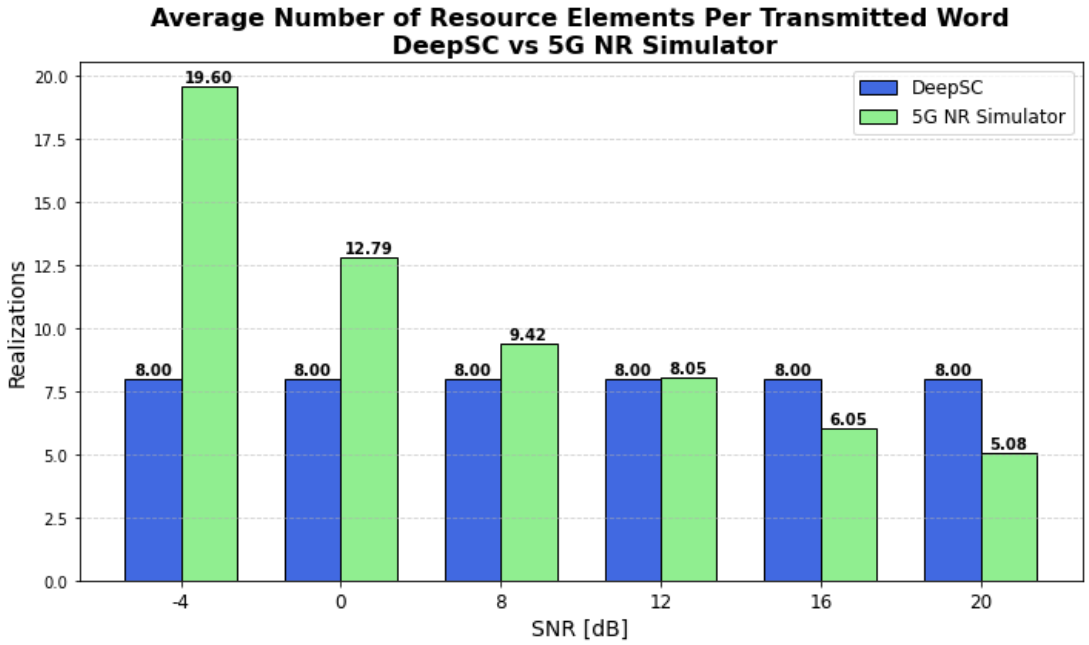


Figure 54: Average number of resource elements per transmitted word

5. Conclusions

5.1 Findings Summary

The experiment carried out for this thesis projectuses the text transmission task as a test caseto compare the performance of a modified version of the DeepSC model-which included a CDL-B channel model, MIMO capabilities and MMSE equalization-with that of a traditional 5G NR-based transmission line. The latter implements a Huffman encoder/decoder and an Adaptive Modulation and Coding mechanism.

The results obtained confirm that the implemented equalization technique and MIMO transmission capabilities enhance the model's performance, as reflected in both performance metrics considered, i.e. BLEU Score and Similarity Score. This improvement can be attributed to several factors:

- the impairments introduced by the channel is mitigated by the effect of the MMSE equalization
- MIMO transmission exploits spatial diversity, is one of the copies of the signal is attenuated on a certain path, it is probable that a copy on another path is less attenuated, hence increasing the probability of receiving a signal with good quality
- MIMO, as expected, also enhances the total channel capacity

Even if the investigated paradigm is promising, the comparison with the traditional 5G NR-based transmission line clearly indicates that the latter outperforms the proposed semantic transmission line.

However, although the semantic communications paradigmwas theorized decades ago, it is still in its early stage of development. This suggests that more refined theories, applications, and models are expected to emerge in the coming years.

Semantic communications is a fascinating research field, with a breakthrough potential. However, at the current development state, it is not necessarily beneficial to completely replace traditional communication systems with semantic communications.

5.2 Next Works

This thesis project provides some insights into the semantic communications, particularly in the context of text transmission use case. However, several areas require further investigation. An overview about the areas where additional research is required is presented in this section.

1. *Additional modifications to the semantic model:*

   Several modifications can be investigated to increase the model performance, starting with number of layers. The proposed model is a relatively simple architecture: additional layers could be introduced to encode the sentences, at the expense of an increased complexity.

   Possibly, the multi-head attention mechanism can be revised too, the number of transformers as well as the number of head can be adjusted, with the goal of enhancing the performance of the model in terms of semantic meaning extraction.
2. *Implementation of precoding techniques into the neural network:*

   For each transmission, the proposed model performs an SVD to obtain the precoding matrix. However, this process is carried out in the channel layer, making it transparent to the neural networks. An alternative approach could involve integrating the precoding process within the neural network layers. This way, after the training phase, the neural network would be capable of executing the precoding operations internally based on the current scenario, eliminating the need for specific precoding operations in the subsequent phases.
3. *Evaluation of more challenging scenarios from a radio point of view:*

   The results show that, even if the 5G NR-based transmission line outperforms the modified DeepSC model, the semantic model utilizes slightly less total bandwidth at low SNR values.. Therefore, it would be interesting to investigate how more challenging scenarios would affect the performance of both systems and whether

the proposed model could surpass the traditional paradigm in terms of bandwidth efficiency.

4. *Evaluation of link adaptation schemes for the semantic communication models:*
   As it is implemented, DeepSC utilizes the same number of symbols regardless of the channel conditions, while the 5G NR simulator takes advantage of the Adaptive Coding Rate mechanism. It would be interesting to investigate the benefits of the introduction of link adaptation schemes for the DeepSC model, changing the number of symbols used to encode a sentence based on the channel quality
5. *Application of the proposed transmission scheme to goal oriented scenarios:*
   This thesis considers text transmission as use case; however, as [25] suggests, semantic communications perform better in goal-oriented systems, where the exchange of data is performed with a specific objective. For example, considering an image classification task, the model could extract and transmit only the semantic features of a specific subject in the image.
   In this kind of scenarios, the semantic paradigm could provide significant benefits: all the surplus information could be excluded from transmission, resulting in savings in both bandwidth and processing workload at the receiver side.
   The considered use case still requires the transmission of all data, i.e., the sentences, therefore the semantic model is not able to fully realize its potential, showing its limitations compared to traditional approaches.
6. *Image and Video transmission use cases:*
   In Chapter 3 the rationale behind the choice of the use case is explained. Text transmission has been chosen due to its interpretability and reduced computational complexity.
   However, this approach does not fully leverage the capabilities of the semantic model. Image and video transmission instead, are more suitable use cases, since they allow for a greater amount of data to be discarded, depending on the goal of the communication.
   Some potential use cases are:
   - Surveillance and security systems, in this case a semantic model could focus on transmitting only the critical segments of videos, for example, where movement or suspicious activities are detected.

- Augmented reality, where a system could transmit only the elements that are strictly relevant to enhance the immersive experience.
- Telemedicine, in the case of medical imaging, a semantic model could prioritise the transmission of regions significant to the diagnosis.

All proposed examples could allow faster communications requiring less bandwidth with respect to traditional communication systems.

Ultimately, the field of semantic communications has great potential, especially for goal-oriented tasks. Rigorously defining the theory behind this new paradigm, exploring new applications, and conducting future research are crucial steps for refining these models and understanding the boundaries of what this paradigm can achieve.

Ringraziamenti

Ringrazio HPC Polito e TIM SpA per avermi fornito la potenza computazionale necessaria allo svolgimento di questa tesi.

Vorrei esprimere la mia sincera gratitudine per il loro prezioso supporto durante questi 8 mesi ai miei relatori accademici, la Prof. Carla Fabiana Chiasserini e il Dott. Corrado Puligheddu, e ai miei relatori aziendali, l'Ing. Roberto Fantini e l'Ing. Elisa Zimaglia, senza la cui guida non avrei potuto svolgere questa tesi.

Un pensiero speciale va poi a tutte le persone che mi sono state accanto durante questo viaggio.

Senza la mia famiglia non avrei conseguito questo risultato e non sarei metà di ciò che sono.

A mia madre e mio padre, i pilastri che mi supportano da 27 anni e un porto sicuro a cui ritornare appena posso (ferie permettendo).

Tanto affetto e riconoscenza vanno ai miei fratelli Eleonora e Lorenzo, questa sarà l'unica volta che non vi chiamerò Ele e Lollo in vita mia, che, anche se mi sopportano da meno tempo rispetto ai miei genitori, sono egualmente importanti per me.

Ogni mio risultato e soddisfazione lo devo anche ai nonni: a nonno Luigi, a nonna Antonietta, a nonno Antonio e a nonna Filomena, grazie per l'affetto che non mi avete mai fatto mancare e per le "strenne" che hanno finanziato i miei gelati lontano da casa.

Ringrazio anche tutti gli zii: Alessandro, Teresa, Rosanna, Coriano, Maria, Raffaele, Sonia, Andrea, Luciana e tutti i cugini: Giuseppe, Luigi, Mario ed Helena; i pranzi e le cene insieme sono ricordi preziosi che conservo.

Un pensiero va ad Alessio, Pietro e Valentino, con cui ho trascorso più serate di quante ne ricordi.

Grazie anche agli amici che non ho citato, a quelli che conosco da una vita e a quelli che ho conosciuto più di recente, la lista sarebbe troppo lunga da fare, ma ci tengo a dirvi che siete importanti.

Ora, guardando al percorso appena concluso, non posso che ringraziare anche i miei colleghi universitari per i due anni trascorsi insieme e per le ansie condivise.

“Joy”, visto che non scrivevo in inglese da un po’, è ciò che mi dà ogni giorno Gon, insieme a svariate fatture dei veterinari, compagno di tante giornate di studio che ha alleviato con la sua presenza.

Oggi non è altro che la fine di un percorso che non avrei saputo affrontare senza Noemi, la persona che più mi ha sostenuto, aiutato e confortato in questi anni, così tanto che non trovo parole per riassumerlo, dal profondo del cuore: GRAZIE.

Bibliography

[1] C. E. S. a. W. Weaver, The Mathematical Theory of Communication, The University of Illinois Press, 1949.

[2] Y. B.-H. e. a. R. Carnap, "An Outline of A Theory of Semantic Information," *Massachusetts Institute of Technology,* vol. RLE Technical Reports, no. 247, 1952.

[3] B. e. al., "Towards a theory of semantic communication," *IEEE Network Science Workshop,* pp. 110-117, Jun. 2011.

[4] L. e. al., "AFSSE: An Interpretable Classifier with Axiomatic Fuzzy Set and Semantic Entropy," *IEEE Transactions on Fuzzy Systems,* Oct. 2019.

[5] e. a. Peng, "A Robust Deep Learning Enabled Semantic Communication System for Text," [Online].

[6] I. J. G. e. al., "Explaining and harnessing adversarial examples," Dec. 2014. [Online]. Available: arXiv preprint, arXiv:1412.6572.

[7] M. e. al., "Adversarial training methods for semi-supervised text classification," *arXiv preprint,* no. arXiv:1605.07725,, 2016.

[8] a. Zhijin Qin, "Semantic Communications: Principles and Challenges," 30 12 2021. [Online]. Available: arXiv:2201.01389.

[9] W. Z. a. H. V. P. J. Liu, "A Rate-Distortion Framework for Characterizing Semantic Information," *IEEE International Symposium on Information Theory (ISIT),* pp. pp. 2894-2899, 2021.

[10] F. C. P. a. W. B. N. Tishby, "The information bottleneck method," Apr. 2000. [Online]. Available: arXiv preprint arXiv:0004057.

[11] M. S. a. E. C. Strinati, “Learning semantics: An opportunity for effective 6G communications,” *arXiv preprint.*

[12] S. R. T. W. a. W. Z. K. Papineni, “BLEU: A method for automatic evaluation of machine translation,” in *Proc. Annual Meeting Assoc. Comput. Linguistics (ACL)*, Philadelphia, 2002.

[13] Z. Q. G. Y. L. a. B.-H. J. H. Xie, “Deep learning enabled semantic communication systems,” *IEEE Trans. Signal Process,* pp. 2663-2675, Apr. 2021.

[14] H. X. a. Z. Qin, “A lite distributed semantic communication system for Internet of Things,” *IEEE J. Sel. Areas Commun.,* pp. 142-153, Jan. 2021.

[15] M. C. K. L. a. K. T. J. Devlin, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proc. North American Chapter of the Assoc. for Comput. Linguistics: Human Language Tech. (NAACL_HLT),* p. 4171–4186, June 2019.

[16] O. K. A. F. Steffen Czolbe, “Semantic similarity metrics for learned image registration,” *arXiv:2104.10051,* 2021.

[17] A. A. a. L. F.-F. J. Johnson, “Perceptual losses for real-time style transfer and super-resolution,” in *Proc. European Conf. Comput.* , Amsterdam , 2016.

[18] P. I. A. A. E. E. S. a. O. W. R. Zhang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recogni*, Salt Lake City, 2018.

[19] Y. S. T. L. C. R. J. W. J. P. B. C. a. Y. W. J. Wang, “Learning fine-grained image similarity with deep ranking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, 2014.

[20] J. P.-A. M. M. B. X. D. W.-F. S. O. A. C. a. Y. B. I. J. Goodfellow, “Generative adversarial netoworks,” in *arXiv preprint arXiv:1406.2661*, 2014.

[21] J. G. B. M. P. H. a. A. P. H. A. W. Rix, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process* , Salt Lake City, 2001.

[22] R. C. H. R. H. a. J. J. C. H. Taal, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," in *IEEE Trans. Audio, Speech, Language Process*, 2011.

[23] C. S. J. B. M. O. R. U. J. P. a. M. K. J. G. Beerends, "Perceptual objective listening quality assessment (POLQA), the third generation ITU-T standard for end-to-end speech quality measurement part I—temporal alignment," in *J. Audio Eng. Soc*, 2013.

[24] J. D. S. D. A. C. E. E. N. C. L. C. C. a. K. S. M. Binkowski, "High Fidelity speech synthesis with adversarial networks," arXiv preprint, September 2019. [Online]. Available: arXiv:1909.11646.

[25] M. G. A. A. B. T. T. M. D. O. A. Mert Kalfaa, "Towards Goal-Oriented Semantic Signal Processing: Applications and Future Challenges," Department of Electrical and Electronics Engineering, Bilkent University, 06800, Ankara, Turkey, 2021.

[26] G. G. Chowdhury, "Natural language processing," *, Annual review of information science and technology ,* p. 51–89, 2003.

[27] D. L. Waltz, "An English language question answering system for a large relational database," *Communications of the ACM 21,* p. 526–539, 1978.

[28] H. J. Z. W. C. F. J. L. Q. You, ",Image captioning with semantic attention," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* p. 4651–4659, 2016.

[29] D. Marcu, “The theory and practice of discourse parsing and summarization,” *,MIT Press,* 2000.

[30] N. P. M. Kountouris, “Semantics-empowered communication for networked intelligent systems,” *arXiv preprint.*

[31] B. Juba, “Universal Semantic Communication,” *Springer Berlin Heidelberg,* 2011.

[32] W. O. B. Z. K. W. X. W. Y. Li, “Scene graph generation from objects, phrases and region captions,” *Proceedings of the IEEE International Conference on Computer Vision,* p. 1261–1270, 2017.

[33] J. Z. J. F. Z. C. Z. Wang, “Knowledge graph embedding by translating on hyperplanes,” *Proceedings of the AAAI Conference on Artificial Intelligence,* p. 1112–1119, 2014.

[34] I. D. C. D. D. R. A. S. K. W. J. J. Carroll, “Implementing the semantic web recommendations,” *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters,* p. 74–83, 2004.

[35] J. D. T. D. J. M. R. Girshick, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 580-587, 2014.

[36] R. Girshick, “Fast R-CNN,” *Proceedings of the IEEE Conference on Computer Vision,* pp. 1440-1448, 2015.

[37] K. H. R. G. J. S. S. Ren, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence 39,* pp. 1137-1149, 2016.

[38] S. D. R. G. A. F. J. Redmon, “You only look once: Uni ed, real-time object detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 779-788, 2016.

[39] A. B. H.-Y. M. L. C.-Y. Wang, "Scaled-yolov4: Scaling cross stage partial network," arXiv:2011.08036, 2021.

[40] R. P. Q. V. L. M. Tan, "E cientdet: Scalable and e cient object detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 10781-10790, 2020.

[41] M. J. R. C. J. Shotton, "Semantic texton forests for image categorization and segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1-8, 2008.

[42] A. F. M. C. T. S. M. F. R. M. A. K. A. B. J. Shotton, "Real-time human pose recognition in parts from single depth images," *CVPR 2011, IEEE,* pp. 1297-1304, 2011.

[43] M. N. S. L. J. Tighe, "Scene parsing with object instances and occlusion ordering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 3748-3755, 2014.

[44] Y. Z. Y. G. S. Hao, "A brief survey on semantic segmentation with deep learning," *Neurocomputing 406,* pp. 302-321, 2020.

[45] P. L. G. M. R. P. Poudel, " Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation," *Reconstruction, Segmentation, and Analysis of Medical Images, Springer,* p. 8394, 2016.

[46] P. F. T. B. O. Ronneberger, "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer,* pp. 234-241, 2015.

[47] C. C. S. C. J. V. P. Luc, "Semantic segmentation using adversarial networks," arXiv, [Online]. Available: arXiv:1611.08408.

[48] G. G. P. D. R. G. K. He, "Mask R-CNN," *Proceedings of the IEEE Conference on Computer Vision,* pp. 2961-2969, 2017.

[49] K. H. R. G. C. R. P. D. A. Kirillov, “Panoptic segmentation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 9404-9413, 2019.

[50] L. A. H. S. G. M. R. S. V. K. S. T. D. J. Donahue, “Long-term recurrent convolutional networks for visual recognition and description,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2625-2634, 2015.

[51] L. F.-F. A. Karpathy, “Deep visual-semantic alignments for generating image descriptions”. *Proceedings of the IEEE preprint arXiv:1410.1090.*

[52] J. B. R. K. K. C. A. C. R. S. R. Z. Y. B. K. Xu, “Show, attend and tell: Neural image caption generation with visual attention,” *International Conference on Machine Learning, PMLR,* pp. 2048-2057, 2015.

[53] A. K. L. F.-F. J. Johnson, “ Densecap: Fully convolutional localization networks for dense captioning,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 4565-4574, 2016.

[54] K. T. J. Y. L.-J. L. L. Yang, “Dense captioning with joint inference and visual context,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2193-2022, 2017.

[55] J. C. I. S. K. e. a. D.-J. Kim, “ Dense relational image captioning via multi-task triple-stream networks”.*arXiv preprint arXiv:2010.03855.*

[56] A. M. W. L. S. Z. G. M. S. N. Aafaq, “Video description: A survey of methods, datasets, and evaluation metrics,” *ACM Computing Surveys (CSUR) 52 ,* pp. 1-37, 2019.

[57] R. K. M. S. L.-J. L. D. S. M. B. L. F.-F. J. Johnson, “Image retrieval using scene graphs,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 3668-3678, 2015.

[58] W. O. B. Z. J. S. C. Z. X. W. Y. Li, "Factorizable net: an e cient subgraph-based framework for scene graph generation," *Proceedings of the European Conference on Computer Vision (ECCV),* pp. 335-351, 2018.

[59] J. L. S. L. D. B. D. P. J. Yang, "Graph R-CNN for scene graph generation," *Proceedings of the European Conference on Computer Vision (ECCV),* pp. 670-685, 2018.

[60] G. C. A. C. A. R. P. L. Y. B. P. Velickovic, "Graph attention networks".*arXiv preprint.*

[61] H. Z. B. W. W. L. W. L. K. Tang, "Learning to compose dynamic tree structures for visual contexts," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 6619-6628, 2019.

[62] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning 8,* pp. 229-256, 1992.

[63] A. L. V. I. M. L. S. D. R. Yu, "Visual relationship detection with internal and external linguistic knowledge distillation," *Proceedings of the IEEE Conference on Computer Vision,* pp. 1974-1982, 2017.

[64] R. K. M. B. L. F.-F. C. Lu, "Visual relationship detection with language priors," *European Conference on Computer Vision, Springer,* pp. 852-869, 2016.

[65] Z. W. P. L. Q. Z. X. H. R. Wang, " Storytelling from an image stream using scene graphs," *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34,* pp. 9185-9192, 2020.

[66] M. K. M. K. M. I. M. M. Malik, "Automatic speech recognition: A survey, Multimedia Tools and Applications," pp. 9411-9457, 2021.

[67] C. P. G. S. R. Collobert, "Wav2Letter: An end-to-end ConvNet-based speech recognition system".*arXiv preprint arXiv:1609.03193.*

[68] e. a. R. Polikar, “ The wavelet tutorial,” 1996.

[69] L. R. R. B. H. Juang, “ Hidden markov models for speech recognition,” *Technometrics 33,* pp. 251-272, 1991.

[70] C.-H. M. L.-S. L. H. Tang, “An initial attempt for phoneme recognition using structured support vector machine (SVM),” *IEEE International Conference on Acoustics, Speech and Signal processing, IEEE ,* pp. 4926-4929, 2010.

[71] Y. Y. H. L. B. Wang, “Attention-based transducer for online speech recognition”.*arXiv preprint arXiv:2005.08497.*

[72] H. L. Y. A. B. S. B. G. O. B. O. S. T. Makino, “ Recurrent neural network transducer for audio-visual speech recognition,” *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU),* pp. 905-912, 2019.

[73] D. G. Tze-Yang Tung, “ DeepWiVe: Deep-Learning-Aided Wireless Video Transmission,” *Information Processing and Communications Lab (IPC-Lab), Imperial College London, UK.*

[74] V. H. C. F. C. d. O. P. S. R. Fernando A. Fardo, “A Formal Evaluation of PSNR as Quality Measurement Parameter for Image Segmentation Algorithms”.*arXiv:1605.07116.*

[75] E. P. S. a. A. C. B. Z. Wang, “Multiscale structural similarity for image quality assessment,” *Seventh Asilomar Conference on Signals, Systems & Computers,* pp. 1398-1402, 2003.

[76] Y. C. S. L. a. H.-S. K. B. Liu, “Deep learning in latent space for video prediction and compression,” *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 701-710, 2021.

[77] D. M. N. J. J. B. S. J. H. G. T. E. Agustsson, “Scale-space flow for end-to-end optimized video compression,” *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 8500-8509, 2020.

[78] Z. Q. G. Y. L. B. J. H. Xie, "Deep Learning Enabled Semantic Communication Systems," arXiv:2006.10685, 2020. [Online]. Available: https://arxiv.org/abs/2006.10685.

[79] P. Foundation, "Pytorch," Pytorch Foundation, [Online]. Available: https://pytorch.org/.

[80] E. P. N.Kalchbrenner, "A convolutional neaural network for modelling sentences," *Proc. Annu. Meeting Assoc. Comput. Linguistics,* pp. 655-665, June 2014.

[81] A. I. a. G. M. L. Atzori, "The Internet of Thingss: a survey," *Computer Networks vol. 54, no. 15,* pp. 2787-2805, October 2010.

[82] A. A.-F. S. S. a. M. G. M. Mohammadi, "Deep Learning for IoT Big Data and Streaming Analytics: A Survery," *IEEE Commun. Surv. Tutorials, vol. 20, no. 4,* pp. 2923-2060, June 2018.

[83] K. O. a. M. D. H. Li, "Learning iot in edge: Deep learning for the Internet of Things with edge computing," *IEEE Network,* vol. 32, no. 1, pp. 96-101, 2018.

[84] W. Z. J. B. Y. L. R. F. E. L. Denton, "Exploiting linear structure within convolutional networks for efficient evaluation," *Proc. Adv. Neural Inf. Process. Syst. (NIPS),* pp. 1269-1277, December 2014.

[85] Z. Q. Huiqiang Xie, "A Lite Distributed Semantic Communication System for Internet of Things," *IEEE Journal on Selected Areas in Communications,* vol. 39, no. 1, pp. 142-153, 2021.

[86] Y. X. Z. L. W. Z. L. F. H. L. C. Tian, "Attention-guided cnn for image denoising," *Neural Netw.,* vol. 124, pp. 117-129, 2020.

[87] [Online]. Available: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum.

[88] W. Z and Q. Z, “Semantic Communication Systems for Speech Transmission,” *EEE Journal on Selected Areas in Communications,* vol. 39, no. 8, pp. 2434-2444, 2021.

[89] E. Vincent, R. Gribonval and C. F´evotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio, Speech, Lang. Process. ,* vol. 14, no. 4, pp. 1462-1469, 2006.

[90] J. G. B. M. P. H. A. P. H. A. W. Rix, “Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs,” Salt Lake City, 2021.

[91] ITU-T, “Perceptual evaluation of speech quality (PESQ): An objective method or end-to-end speech quality assessment of narrow-band telephone networks and speech codes,” 2001.

[92] J. H. B. D. J. C. Shailesh Chaudhari, “arXiv:1408.6587,” [Online]. Available: https://arxiv.org/abs/1408.6587.

[93] ETSI, “TR 138 901 V16.1.0,” European Telecommunication Standard Institute, 2020.

[94] 3. G. P. Project, “TS 38.214,” 2021. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.214/. [Accessed 10 October 2024].

[95] D. A. Huffman, “A Method for the Construction of Minimum-Redundancy Codes,” *Proeedings of the IRE,* vol. 40, no. 9, pp. 1098-1101, 1952.

[96] O. S. J. K. S. Park, End-to-end fast training of communication links withoutachannelmodelviaonlinemeta-learning, arXiv:2003.01479, 2020.

[97] R12-SG05, “IMT Vision - "Framework and overall objectives of the future development of IMT for 2020 and beyond",” 2015. [Online]. Available: https://www.itu.int/rec/R-REC-M.2083-0-201509-I/en.

[98] E. Dahlman, S. Parkvall and J. Skold, 5G NR: The Next Generation Wireless Access Technology, Academic Press, 2018.

[99] Erdal Arıkan, Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels, 2009.

[100] K. Niu, K. Chen, J. Lin and Q. T. Zhang, Polar Codes: Primary Concepts and Practical Decoding Algorithms, IEEE Communications Magazine, July 2014.

[101] H. Vangala, Y. Hong, E. Viterbo and A. Monash University, "A Practical Introduction to Polar Codes," 23 February 2016. [Online]. Available: is.gd/polarcodes.

[102] I. Tal and A. Vardy, List-Decoding of Polar Codes, 9500 Gilman Drive, La Jolla, CA 92093, USA: University of California San Diego.

[103] I. Tal and A. Vardy, List Decoding of Polar Codes, IEEE, 2015.

[104] A. F. J. Redmon, " YOLO9000: Better, faster, stronger," *Proceedings of the IEEE Conference on Computer VIsion and Pattern Recognition,* pp. 7263-7271.