Integration of contextual knowledge in deep Learning modeling for vision-based scene analysis

Fatima Ezzahra Benkirane

THÈSE DE DOCTORAT

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

FATIMA EZZAHRA BENKIRANE

Integration of Environment Contextual Knowledge in Deep Learning Modeling for Vision-based Scene Analysis

Thèse présentée et soutenue à UTBM Montbéliard, le 28 Février 2024

Composition du Jury :

Mme. ABEL MARIE-HÉLÈNE	Professeur des universités à l'UTC	Examinatrice (*Présidente du jury*)
Mme. CHAMBON SYLVIE	Maître de conférences à l'IRIT	Rapportrice
M. TALEB-AHMED ABDELMALIK	Professeur des universités à l'UPHF	Rapporteur
M. HILAIRE VINCENT	Professeur des universités à l'UTBM	Directeur de thèse
M. RUICHEK YASSINE	Professeur des universités à l'UTBM	Codirecteur de thèse
M. CROMBEZ NATHAN	Maître de conférences à l'UTBM	Coencadrant de thèse

N°

ACKNOWLEDGEMENTS

Tout d'abord, je tiens à exprimer ma reconnaissance particulière au jury de soutenance d'avoir accepté d'évaluer mon travail. Je remercie les professeurs Sylvie Chambon et Abdelmalik Taleb-Ahmed pour avoir consacré du temps à la lecture et à l'évaluation de mon manuscrit de thèse. De plus, je tiens à remercier Professeur Marie-Hélène Abel d'avoir accepté d'être examinatrice lors de la soutenance de ma thèse.

Je tiens à exprimer ma profonde gratitude envers mon directeur de thèse Pr. Vincent Hilaire dont le soutien constant a été un pilier essentiel tout au long de ce parcours de recherche. Je le remercie pour ses conseils éclairés et ses directives avisées qui ont guidé mes travaux scientifiques. Son engagement envers l'excellence académique et sa disponibilité constante ont grandement contribué à la réussite de ce projet de thèse. Merci infiniment pour cette collaboration fructueuse et cette aventure intellectuelle enrichissante.

Je tiens également à remercier infiniment mon codirecteur de thèse, Pr. Yassine Ruichek, pour le soutien, l'encouragement et l'orientation stratégique qu'il m'a accordés dans ce parcours. Grâce à ses conseils, son expertise et sa vision stratégique, ma thèse de doctorat a pris forme, aboutissant à des contributions et publications scientifiques dont je suis fière aujourd'hui. Je remercie Pr. Ruichek pour ses efforts afin de créer un environnement propice à l'apprentissage et à l'épanouissement intellectuel. Son engagement a créé un cadre stimulant et collaboratif, favorisant la créativité et la progression intellectuelle au sein de l'équipe de recherche. Tous mes remerciements d'avoir mis à ma disposition toutes les ressources nécessaires, notamment les machines et ressources de calcul de pointe, qui ont contribué à l'avancement rapide et la qualité de mes recherches. Merci pour la confiance, les valeurs et le savoir qui ont été des catalyseurs essentiels de ma réussite académique.

Je tiens à exprimer ma reconnaissance envers mon encadrant de thèse Dr. Nathan Crombez, pour ces trois années d'apprentissage enrichissantes. Son encadrement de qualité et ses remarques fructueuses ont constamment été formateurs pour moi et ont grandement contribué à l'enrichissement de mes travaux scientifiques. Je le remercie pour sa disponibilité dès le premier jour et de m'avoir initié et appris les fondamentaux de la recherche scientifique. Je suis reconnaissante de son soutien constant, de sa confiance, et de son encouragement à donner le meilleur de moi-même, ce qui a été un pilier essentiel pour l'avancement de ma carrière en tant que chercheuse. Les con-

seils techniques, stratégiques, scientifiques et organisationnels qu'il m'a transmis sont d'une très grande valeur. Je souhaite lui exprimer ma reconnaissance pour avoir transmis l'importance du sens du détail, des réflexions approfondies par rapport aux idées et aux motivations scientifiques, autant d'éléments fondamentaux pour devenir un bon chercheur. Son engagement et son intégrité envers ce projet ont été exemplaires, et je suis convaincue que la réussite de ce projet de thèse est en grande partie grâce à lui. Merci pour cette belle aventure.

Je remercie mes collègues, les membres du laboratoire CIAD, pour l'environnement de travail stimulant et collaboratif. Merci pour les échanges et les moments conviviaux que nous avons partagés. Je suis fière d'appartenir à une équipe de valeurs avec des membres engagés. Je remercie également la structure professorale du laboratoire CIAD pour les efforts continus en vue du développement et du rayonnement du CIAD.

Finalement, je souhaite exprimer ma reconnaissance envers ma famille et mes amis qui ont été une source de soutien moral et émotionnel tout au long de mon parcours. Leurs encouragements m'ont motivé à persévérer dans les moments difficiles. Leurs paroles réconfortantes et leur présence ont été des boucliers contre les défis auxquels j'ai dû faire face, me donnant la force nécessaire pour surmonter les obstacles. Merci du fond du cœur.

CONTENTS

1

INTRODUCTION

1.1/ AUTONOMOUS VEHICLES CONTEXT

Autonomous vehicles (AV), also known as self-driving cars, have become an important breakthrough in today transportation field. Since the last century, this topic has been one of the main research topics in both industry and academia regarding the advantages it can provide [1]. First, it can improve safety by reducing the number of accidents caused by human errors. Second, it reduces traffic congestion and travel time while increasing the efficiency of the transportation system [2]. In addition, it enables better comfort and security for drivers and passengers. Finally, it has a social inclusion impact because it allows mobility for everyone, including the underlay and handicapped people [3].

1.1.1/ AUTONOMOUS DRIVING LEVELS

An AV may only be considered as “autonomous" when it can handle all dynamic driving tasks in an environment [4]. According to the Society of Automotive Engineers (SAE) [5], there are five different levels of automation in vehicles, ranging from level 0 (no automation) to level 5 (complete automation) represented in Figure 1.1. At level 0, the driver controls every part of the car, whereas at level 1, multiple forms of driver assistance, such as adaptive cruise control, are used. Level 2 includes partial automation, where the vehicle can perform some tasks, but driver assistance is still required. In level 3 conditional automation is introduced, where the vehicle can handle the majority of assignments but may need human assistance occasionally. High automation is achieved at level 4, allowing the vehicle to handle multiple situations without human intervention. Finally, level 5 is the symbol of complete automation, without human involvement. In this context, there are already some Advanced Driver Assistance Systems (ADAS) that are currently integrated into commercial vehicles, such as adaptive cruise control, Lane Keeping Assist Systems (LKAS), and Automatic Emergency Braking (AEB). These ADAS functions correspond to level 2 automation. Furthermore, some taxis and low-speed shuttles are examples of

level 4 vehicles that already exist, but they are trial projects that are only allowed to run in specific locations and at specific speeds while being continuously tested to demonstrate their effectiveness. However, level 5 vehicles, which to the best of our knowledge do not yet exist, are claimed to have no operational limitations and to be able to navigate anywhere at any time according to the SAE [5].

To reach high levels of automation, a strong and structured AV system is required. This system must properly observe, detect, and understand its surroundings to make decisions that are similar to those of a human. In the following section, we present a general overview of the AV systems' architecture.

1.1.2/ AUTONOMOUS VEHICLE SYSTEM ARCHITECTURE

A self-driving vehicle must perform four fundamental tasks to operate without human assistance: localization and mapping, perception, path planning, and control [6]. Localization and mapping tasks involve the construction of a map representing the vehicle environment and maintaining continuous awareness of the vehicle location in relation to that map. Perception is considered the main component of an intelligent system and aims at modeling the environment. To perceive its surroundings effectively, the vehicle should perform several key tasks. First, the vehicle perception system uses a set of onboard sensors, such as cameras, lidar, radar, etc., to obtain raw data from the environment. The next crucial step involves the transformation of raw sensor data into useful information. The raw data collected by these sensors are processed by advanced algorithms and computer vision techniques. By interpreting, filtering, and processing the data, these algorithms enable the vehicle to detect, identify, and determine the depth, direction, as well as position of the surrounding objects of the environment. Path planning combines the outcomes of perception and localization to choose the most secure and effective path for the AV, considering all potential obstacles [7]. Finally, the control element outputs the acceleration, torque, steering angle values, and other actions required for the vehicle to follow the selected path [8].

Every component of the AV systems' architecture contributes efficiently to ensure the smooth functioning and safety of the system. Perception, localization, path planning, and control collectively collaborate to make the vehicle fully autonomous. However, from our perspective, among these crucial elements, perception stands as the heart and core of the entire system. This is because it plays an important role in interpreting the surroundings, enabling the vehicle to make informed decisions and navigate complex environments. In this context, it has been demonstrated that environmental perception performance directly affects the performance of autonomous driving technology [9]. Without an accurate and precise perception system, the AV cannot make informed decisions or take

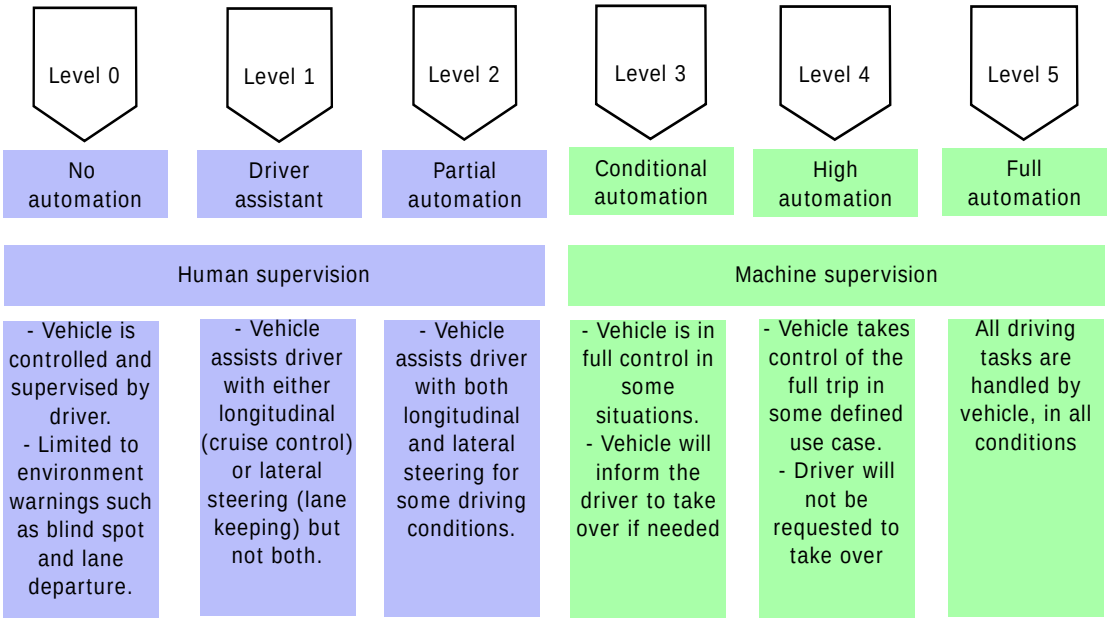


Figure 1.1: Autonomous driving levels

appropriate actions. All subsequent processes are built based on a robust perception system.

Within the AV perception system, computer vision tasks refer to specific techniques and

algorithms used to extract useful information from the visual data collected from the sensors. These tasks are essential to analyze visual information, detect objects and patterns, and ultimately help the vehicle understand its surroundings. In this context, the development of computer vision has progressed from basic image processing methods in the $1960s$ to Deep Learning (DL) models able to identify and analyze complicated visual patterns nowadays. In the next section, we will provide an overview of the evolution of computer vision techniques along with an analysis of current challenges in the field.

1.2/ Computer vision: Evolution, advantages and challenges

To understand the environment, humans rely primarily on their senses. We can recognize individuals, identify objects, and fully understand the emotions of others, thanks to the abilities of our visual system. For decades, scientists have been fascinated by automating and representing this complex process, which led to the development of the well-known computer vision [10]. This field considerably transformed the way machines detect and interpret visual data. Over the years, it has developed through two main phases. This journey began with traditional or feature-based techniques to reach the era of DL approaches. In computer vision, features refer to patterns and characteristics within an image that enable machines to understand and analyze visual data. Feature descriptors, on the other hand, are techniques and algorithms used to encode these distinct features in a suitable way for computer vision tasks.

Traditional computer vision techniques relied on feature descriptor methods such as Scale Invariant Feature Transform (SIFT) [11], Speeded Up Robust Features (SURF) [12], and Features from Accelerated Segment Test (FAST) [13], to perform various computer vision tasks. In traditional approaches, a critical step involves feature extraction, especially in tasks like image classification or object detection. Various techniques, including edge detection and threshold segmentation, were used for feature extraction. These techniques aim to recognize and identify visual features present within an image. Let us take the example of an object detection task. For each detected object class or category, a distinct representation is generated using the extracted features. A term from Natural Language Processing (NLP), the “bag-of-words" model [14], was frequently used to describe this structure. When used in this context, it denotes that image features were viewed as a set of visual words that collectively described an object or scene.

Hand-crafted methods have multiple advantages, mainly in efficiency and simplicity. These algorithms often need few lines of code and processing resources, which makes them particularly useful in resource-constrained contexts. Additionally, they provide a

high degree of transparency, which makes it simple for users to adjust and fine-tune their parameters to suit different scenarios. However, a significant challenge associated with traditional methods lies in the need to manually select discriminative features from images. As the number of classes increases and when dealing with huge datasets, this task becomes more challenging. It heavily depends on the perspective of computer vision experts and involves an extensive process to define the adequate features [12]. Although traditional methods could be sufficient for smaller datasets and easier applications, they are often limited when it comes to more challenging applications, such as AVs that operate in urban environments. In these complex scenarios, the need for methods and algorithms that can handle challenging environments becomes apparent and important. In this context, DL offered a more powerful solution with generalization capability to address the challenges mentioned above. This involves introducing the theory of “end-to-end" learning, in which the machine is provided with a dataset that includes input samples paired with corresponding annotations. As a result, a DL model is trained on the input data, where Deep Neural Networks (DNNs) identify the underlying patterns in the input and automatically define the most descriptive features for each instance. In this case, a significant part of the computer vision task process is performed automatically by the DL model, reducing the need for human intervention, as shown in Figure 1.2.

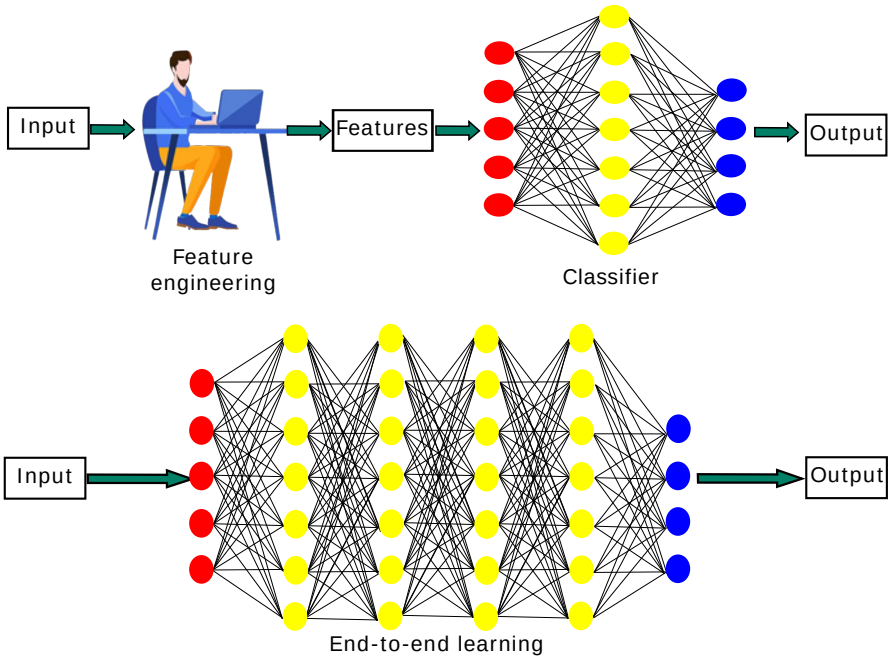


Figure 1.2: Traditional computer vision workflow vs. deep learning workflow

Compared to traditional computer vision techniques, DL approaches offer distinct advantages. It enables computer vision experts to achieve higher accuracy on multiple tasks. The key difference is that DNNs are trained rather than explicitly programmed. This means that DL applications often require less expert analysis and fine-tuning, mak-

ing them more adaptable to various scenarios. In addition, DL approaches provide exceptional flexibility. For example, Convolutional Neural Network (CNN) [15] models and frameworks can be re-trained with custom datasets for specific use cases. This stands in contrast to traditional algorithms, which are often more domain-specific and less flexible in accommodating different applications [16].

Despite these advantages, there are also some challenges related to DL methods. While the most recent approaches can significantly improve the accuracy, there is a cost associated with this progress. Such improved performance requires billions of extra mathematical calculations, increasing the need for computational resources. Therefore, Artificial Intelligence (AI) developers need to have available specialized hardware, including Graphics Processing Units (GPUs) [17] and Tensor Processing Units (TPUs) [18] for training. Moreover, DL approaches are heavily dependent on large datasets. Consider the example of some famous computer vision datasets such as the PASCAL VOC dataset [19], which has 500k images covering 20 object categories, or ImageNet [20], which has 1.5 million images covering 1000 object categories. This means that when large datasets or powerful computing resources are not available, DL methods are not the best choice.

Another limitation of DL approaches is their total dependence on visual features and data characteristics during training. When considering that computer vision tasks aim to emulate human understanding and interaction with the environment, this total dependence covers just the way humans “see" their surroundings. However, to fully understand their environment, make decisions, and take actions, humans rely not only on what they “see" but also on what they “know", i.e., their accumulated and pre-acquired knowledge. Although recent advances have shown that DL models can implicitly learn some rules and basic knowledge through neural connections during training, this implicit knowledge acquisition remains imprecise, time-consuming, and strongly dependent on training data. To achieve the highest levels of accuracy and closely emulate human behavior, it is necessary to explicitly incorporate knowledge into DNNs. This need is particularly important in sensitive applications like autonomous driving, where the priority lies in ensuring a high level of safety.

1.3/ Role of knowledge integration in perception for autonomous navigation

Human perception is a dynamic system that allows us to understand the world. It operates through our senses which continually gather an extensive array of information from the surrounding environment. Consider, for instance, the act of driving, human perception here involves the eyes that capture the road, assess traffic, and monitor pedestrians

[21; 22]. This input is then processed by our brain, where it is integrated, interpreted, and combined with our existing knowledge and past experiences. Human perception goes beyond recognizing visual cues; it involves context and a deep understanding of the world. This includes the ability to identify objects even in unusual situations, such as when visibility is low due to adverse weather conditions or obscured by other factors. In such situations, humans combine their pre-acquired knowledge with limited visual cues to navigate safely and make informed decisions.

On the other hand, machine perception, as seen in the actual AI and DL models, primarily relies on data-driven approaches. Machines can process massive amounts of data and recognize patterns efficiently but often struggle to understand the context, extract meaningful insights, or incorporate prior knowledge. Although AI systems can succeed at specific tasks, they typically need the holistic understanding and generalization capability linked to human perception. The gap between machine and human perception highlights the need for AI research and development to move beyond data-driven approaches towards the incorporation of contextual understanding and explicit knowledge, to bridge this gap and bring machines closer to human-like perception.

As illustrated in Figure 1.3, the main difference between human and machine vision becomes more apparent during the interpretation phase. In human vision, this step is dependent on the human brain, which combines knowledge with visual information. However, machine perception focuses only on processing visual features. Although a well-trained DL model does improve accuracy and reduce errors, this enhancement can be further optimized by incorporating additional knowledge into the model.

Certainly, bridging the gap between human and machine perception in computer vision involves answering some fundamental questions. First, we must define the most effective way to combine knowledge with visual cues. This involves developing robust frameworks and algorithms that combine domain-specific knowledge with raw visual data. Second, we need to define the specific knowledge that should be integrated into each computer vision task. Different tasks may require different forms of knowledge. For example, recognizing emotions in facial expressions might require psychological insight, while estimating depth might benefit from some geometric and semantic knowledge. Finally, it is crucial to decide where this knowledge integration should occur in the DL process. Should it be incorporated at the data input stage, during the model training phase, or directly in the architecture of the DNNs? Striking the right balance is important for optimizing the performance of DL models in various computer vision tasks. The strategy to reduce the gap between human and machine perception in computer vision lies in answering three essential questions, “how", “what", and “where" when it comes to integrating knowledge into the DL process.

The concept of combining Machine Learning (ML) or DL with domain-specific knowledge

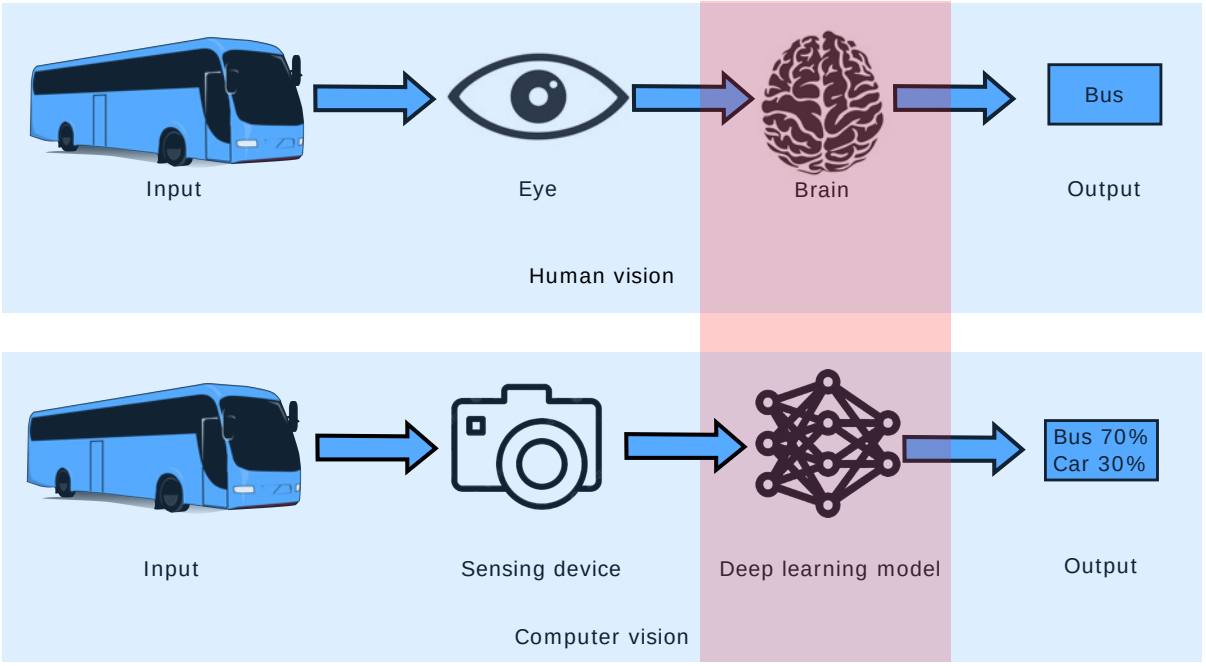


Figure 1.3: Comparison of human vision and computer vision using a classification example. The red block illustrates the gap between the two visions.

is often referred to as hybrid AI. This field represents an intersection between traditional Knowledge-Based Systems (KBS) and data-driven approaches. In hybrid AI, the goal is to take advantage of both knowledge-based techniques and data-driven approaches to improve overall performance and understanding of AI systems. Hybrid AI has found considerable success in various fields, especially in physics [23], hydrology [24] and environment [25]. In these domains, hybrid AI has been instrumental in solving complex problems by combining the deep knowledge of domain experts with the data processing capability of AI systems. However, its adoption in more challenging applications, such as computer vision and, more specifically, AVs, has been somewhat limited. In the context of AVs, the integration of knowledge in perception-decision-making processes remains a relatively unexplored border. While it has promising potential to improve the safety and reliability of autonomous driving systems, it also presents unique challenges, such as knowledge definition and integration, which require further research and development.

1.4/ Problem Formulation

It is worth noting that in computer vision, some approaches have been proposed to combine DNNs with KBS. Many of these approaches use knowledge bases or ontologies as references to validate the results generated by DNNs [26; 27; 28]. While this verification and validation approach does contribute to enhancing the robustness and reliability

of computer vision systems, it primarily focuses on post-processing and confirming the output rather than effectively integrating knowledge into the DL training process. The challenge lies in going beyond the verification step and finding innovative approaches to integrate prior knowledge into the training process of DNNs. Achieving this integration could lead to a more context-aware and adaptable computer vision model.

Additionally, it is important to note that while the approaches mentioned earlier have been proposed effectively for various computer vision tasks, their application within the context of autonomous driving remains limited because of the related challenges. One possible reason is the dynamic nature of urban environments, which presents a unique set of complexities that make it difficult to define and integrate the relevant knowledge into DNNs. In the context of urban environments, this includes changing traffic patterns, unpredictable pedestrian behavior, changing weather conditions, and a multitude of possible road scenarios. These dynamic factors make it complex to pre-define the precise knowledge that should be integrated into DL models.

1.5/ Fundamental concepts of Deep Neural Networks

Neural networks were created to imitate the human nervous system in machine learning, guiding artificial neurons in a manner similar to how human neurons work. Neural networks are used for a wide range of tasks, including image and speech recognition, natural language processing, and various other applications in artificial intelligence. The term “deep learning" is often used when referring to neural networks with multiple hidden layers, which have proven highly effective in capturing complex patterns in data.

In a single-layer network, a set of inputs is directly linked to an output through a generalized form of a linear function. This fundamental form of a neural network is commonly known as a perceptron that is shown in Figure 1.4. The perceptron is based on the concept of weighted sum of inputs followed by an activation function. The perceptron model is the set of weights to be optimized through a learning process considering a training observation of form (X, y) where X is the features vector $[x_1, x_2, \ldots, x_m]$ with a length of m and y is its annotation. The m features are linked to the node through a set of weights $W = [w_1, \ldots, w_m]$ calculating the weighted sum that can include a bias b_0 to emphasis the non-linearity then applying an activation function, hence defining the output value of the perceptron as delighted in Eq. 1.1.

$$\hat{y} = f(X, W) = g(b_0 + \sum_{i=1}^{m} w_i.x_i) \tag{1.1}$$

In the case of multi-layer neural networks, neurons are organized in a layered structure

Figure 1.4: Representation of a perceptron

where input and output layers are positioned on either side of one or more hidden layers. This structured arrangement of layers in the neural network is also called a feed-forward network. The optimization process of a perceptron, that is generalized to neural networks, relies on finding the optimal set of weights to minimize the prediction error between $\hat{y}$ and the annotation y using a loss function such as $L1$ or Mean Squared Error (MSE) errors overall the training set D with n observations forming the objective function represented in Eq. 1.2.

$$\hat{W}^* = argmin_W \frac{1}{n} \sum_{i=1}^{n} loss(f(X^i; W), y^i) \tag{1.2}$$

While the objective function is defined over the entire training data D, the optimization algorithm of neural networks, referred to as backpropagation algorithm, relies on feeding each input data instance X into the network one by one (or in small batches) to create the prediction $\hat{y}$. The next step consists of updating the set of weights as defined in Eq. 1.3 based on the gradient of the error value $E(X) = loss(\hat{y}, y)$. The parameter lr regulates the learning rate of the neural network that can be seen as a freedom factor influencing how quickly or slowly the model converges to the optimal set of parameters. High learning rate values can lead the model to overshoot the minimum of the loss function and diverge instead of converging. On the other hand, very low learning rates can result in slow convergence, which might require a large number of iterations to reach the optimal weights combination.

$$W_{new} = W_{Current} - lr. \frac{\partial E(x)}{\partial W_{Current}} \tag{1.3}$$

1.5.1/ Convolutional Neural Networks

The learning capacity of perceptron in its basic form motivated the machine learning community to extend the perceptron concepts and adapt them for more complex task such as natural language processing with Recurrent Neural Networks (RNNs), time-series pattern recognition with Long Short-Term Memory (LSTM) networks, and also Convolutional Neural Networks (CNNs) that are widely serving computer vision in which this thesis is interested.

In contrast, CNNs are more complex architectures designed for tasks such as image recognition and scene visual perceptions. While the basic perceptron is not directly used in CNNs, the fundamental concepts of neurons, weights, and activation functions are

extended and adapted in CNNs for more sophisticated operations. The main extensions can be summarized in the following:

- **Neurons and Activation Functions:** In a perceptron, each input is connected to a neuron, and the weighted sum is passed through an activation function to produce the output. In a CNN, the basic processing unit is still a neuron, but these neurons are organized in layers, and the activation functions used are often nonlinear (e.g., Rectified Linear Unit (ReLU)).

- **Weights and Convolution:** In a perceptron, each input has an associated weight. In CNNs, weights are used in convolutional layers to detect features in localized regions of the input. Convolutional operations involve sliding a filter (also called a kernel) over the input to perform element-wise multiplications and summing the results. This process replace the weighted sum in a perceptron.

- **Pooling layers:** CNNs often include pooling layers to downsample the spatial dimensions of the input. Max pooling, for example, selects the maximum value from a set of values in a region. This operation helps reduce the computational load and focuses on the most important features.

- **Multiple Layers and Hierarchical Features:** CNNs typically consist of multiple convolutional and pooling layers arranged hierarchically. Each layer extracts higher-level features from the input. This hierarchical feature extraction is similar to the way multiple layers in a neural network process information in a more abstract manner.

- **Fully Connected Layers:** While convolutional and pooling layers capture spatial features, fully connected layers at the end of a CNN combine these features for classification. These layers are like the structure of a traditional neural network, with neurons connected to all neurons in the previous layer, similar to a perceptron architecture.

- **Transposed Convolution Layers:** Transposed convolutions are often employed in the decoder part of CNNs to generate high-resolution feature maps or images from the encoded spatial features from earlier convolution and pooling layers. They are crucial in architectures like U-Net and various generative models where upsampling is needed to produce detailed outputs.

Similar to the perceptron backpropagation learning algorithm, CNNs are learned in the same way but they require more iterations and training samples regarding their set of weights that can reach millions of trainable parameters even with basic architectures. The computer vision experienced the proposal of different architectures of CNNs defining how the convolution-based blocks (convolution, pooling and normalization layers) are linked

to each other. Three architectures are commonly used in the literature that are plain, residual, and sparse connections as illustrated in Figure 1.5.

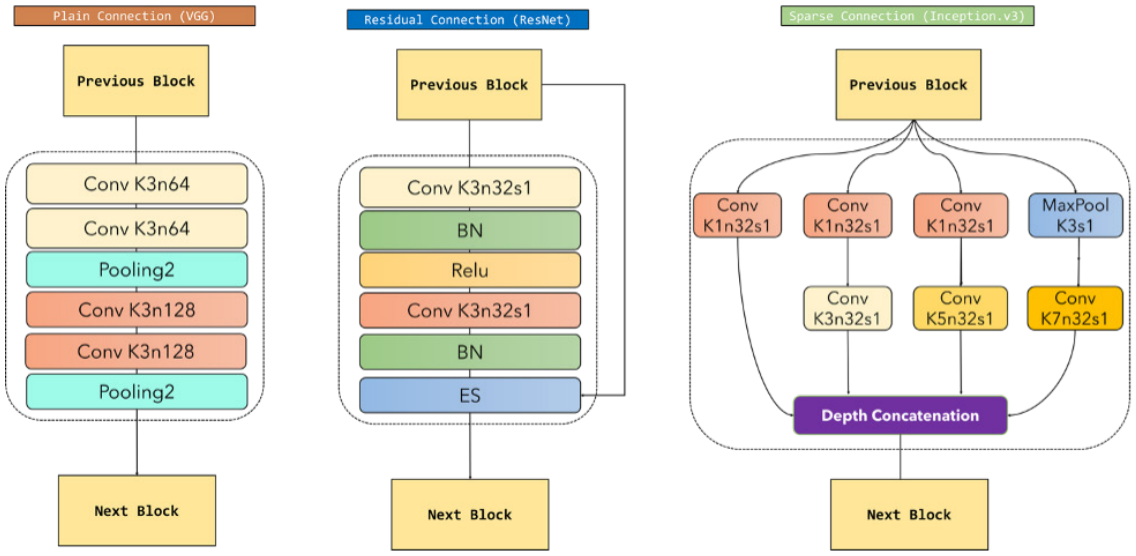


Figure 1.5: Representation of CNNs architectures

In plain or standard connections, each layer output is directly connected to the subsequent layer input. The convolutional layers typically use regular convolution operations to process input data. The information flows sequentially from one layer to the next. This is the basic and most common type of connection in CNNs and is used in AlexNet [29] and VGG [30] models. Residual connections, introduced in the context of Residual Neural Networks (ResNets) [31], involve adding a shortcut or skip connection that bypasses one or more layers. This allows the network to learn residual functions, making it easier to train deep networks. Residual connections help mitigate the vanishing gradient problem and enable optimal and fast training of very deep networks. DenseNet models [32] also adopted the residual connections but using the channel-wise concatenation instead of the sum operation. However, this concatenation results in high computational cost compared to ResNets. Sparse connections employ the definition of a set of convolution with different configurations (kernel, stride, padding) in the same level which can help the model to capture different features. The computed responses are then concatenated and fed to the next block until reaching a latent space. Inception models [33] are the well-known architectures with sparse connections.

Scene visual perception, which is the targeted topic is this thesis, relies CNN encoder-decoder architecture leveraging the power of deep learning to process and interpret complex visual scenes. The encoder-decoder architecture is commonly used for tasks like semantic segmentation, depth estimation, and scene parsing. The encoder is a set of convolution and pooling layers that extract relevant features from the input image, which reflects the scene in our case. Moreover, the encoder tries also to perform the dimension-

ality reduction by keeping only the discriminant features and downsampling the spatial resolution of the input image. On the other hand, the decoder accepts the encoded features and reconstructs their respective prediction through transposed convolutions and upsampling layers. The backpropagation is adjusting the weights of the encoder and the decoder to minimize the error between the prediction and the ground truth. Therefore, the encoded features will be adapted to the intended task. The last layer of the decoder (decoder head) should be adapted to fit the nature of the prediction (depth estimation, semantic segmentation, instance segmentation, etc). Residual-based convolution blocks are often used in the encoder parts to ensure discriminant features extraction with low computational complexity.

1.6/ Fundamental concepts of Knowledge-Based Systems

1.6.1/ Definition and general concepts

Knowledge-based systems (KBS) refer to computer systems that leverage explicit knowledge and reasoning mechanisms to make decisions. These systems are designed to capture, represent, and apply human expertise in a specific domain. Unlike traditional systems that rely on explicit programming, KBS use a knowledge base, which contains facts, concepts and rules to draw inferences and make informed decisions. The knowledge base is typically created by domain experts to improve other systems performance. KBS are commonly employed in areas where human expertise is crucial, such as medicine and engineering. Building a KBS involves several steps, ranging from defining the problem domain to implementing the whole system. The general outline of the key steps in developing a KBS is described as follows:

1. **Define the Problem Domain:** In the initial phase, it is crucial to precisely identify the specific problem or task that the KBS is intended to address. This involves setting clear boundaries and defining the scope of the problem domain. A well-defined problem statement ensures that the subsequent development process remains focused and results in a system that effectively meets its intended objectives.
2. **Knowledge acquisition:** Following the definition of the problem domain, knowledge acquisition involves gathering expertise from domain experts. The goal is to extract relevant facts, rules and other essential information related to the problem domain. This knowledge forms the basis for the subsequent construction of the general KBS.
3. **Knowledge representation:** Once the important and necessary information has

been gathered, it is essential to choose an appropriate knowledge representation scheme. This could involve employing rules, frames, semantic networks, or ontologies to structure and organize the knowledge. The selected representation scheme should be well-suited to the nature of the problem domain and facilitate effective reasoning within the system.

4. **Define the inference mechanism:** The inference mechanism is a critical component that defines how the system processes and draws conclusions from the constructed knowledge base. This step involves selecting or designing an inference engine dedicated to the specific needs of the KBS. Additionally, the rules, algorithms, or reasoning methods that guide the inference process need to be explicitly specified. This ensures that the system can effectively take advantage from the knowledge to make informed decisions.

1.6.2/ Examples of knowledge representation

As mentioned in the previous section, the representation of knowledge is a fundamental aspect of building KBS, and various methodologies have evolved to effectively capture and organize information. Among these, graph networks [34], ontologies [35] or general knowledge bases stand out as prominent forms of knowledge representation. Graph networks use graph structures to represent entities and relationships to organize information. Specifically, it is based on nodes and edges structures to represent the entities and the relationships. Ontologies, meanwhile, provide a formal and explicit specification of a shared conceptualization, defining the vocabulary and relationships within a particular domain. Finally, a knowledge base is essentially a repository of structured information, including facts, rules, and other pieces of knowledge relevant to a particular domain. It acts as a centralized source from which a system can draw information to conduct reasoning and make informed decisions. Each of these approaches serves distinct purposes, offering unique advantages in modeling knowledge and facilitating reasoning. This diversity in knowledge representation methods ensures adaptability to different domains, providing a foundation for developing sophisticated KBS that can interpret, reason, and make informed decisions.

1.6.2.1/ Ontologies

In this thesis, we are focusing on using ontologies as a key element of the proposed KBS in our first contribution. In the aim of clarification, we introduce in this section some basic concepts about ontologies.

Ontologies are a powerful knowledge representation technique that structures information

in a way that captures the relationships and dependencies within a domain. An ontology is essentially a formal, explicit specification of a shared conceptualization, providing a common understanding of a domain vocabulary and relationships. The key concepts and definitions related to ontologies are described as follows:

- **Classes:** They represent categories or sets of objects that share common characteristics. For example, in an ontology representing urban driving environment, “TrafficSign" could be a class that includes instances such as “RedSigne" and “GreenSigne".
- **Individuals:** They represent specific instances of classes, representing particular objects or entities in the domain. For example, an individual instance could be “VehicleX" belonging to the class “Vehicle"
- **Object properties:** They represent relationships between individuals or between individuals and classes. They connect instances of classes in the ontology, indicating some form of object-to-object relationship. For example, an object property “isLocatedOn" linking the class “Person" to the class “Sidewalk" could be used to represent the relationship that a person is on the sidewalk.
- **Data properties:** They are relationships that link individuals to data values, such as strings, numbers, or dates. These properties are used to attribute specific data values to instances. For example, a data property “hasAverageHeight" can link the class “Person" to a numeric value, representing the average height of a person.
- **Rules:** A rule is a statement that defines a relationship or constraint between different entities in the ontology. For example, a rule could be “If the traffic light is red, then the vehicle must come to a complete stop".
- **Reasoning:** Reasoning involves the logical processes used by the system to extract new information from existing knowledge. For example, reasoning could involve defining the optimal route for an autonomous vehicle based on real-time traffic conditions and vehicle capabilities. In the context of ontologies, reasoners play an important role in making logical inferences. Examples of reasoners include Pellet [36], HermiT [37], RacerPro [38], Fact++ [39] and Jena Reasoner [40].

Ontology encoding To encode ontologies, various languages and frameworks are available, with different complexity levels. One widely used language is the Resource Description Framework (RDF) [41], which uses triples that consist of three parts: the subject, the predicate, and the object to represent relationships and concepts. RDF Schema (RDFS) [42] extends RDF by providing a basic vocabulary for defining ontologies. Additionally, the Web Ontology Language (OWL) [43] offers a more expressive and compre-

hensive means of encoding ontologies. OWL allows the specification of classes, properties, and relationships with varying degrees of complexity, making it suitable for a wide range of applications. Common serialization formats for RDF and OWL include RDF/XML [44], Turtle [45], and JSON-LD [46]. The choice of encoding language depends on many factors such as the level of expressiveness needed, the complexity of the domain, and interoperability requirements with existing systems. Each language has its strengths and trade-offs, providing knwoledge experts with many options for encoding the ontology to represent knowledge in a machine-readable and format.

Ontology querying Ontology querying refers to the process of retrieving information from an ontology or using specific queries. Querying involves expressing questions or requests for information in a way that the system can understand and interpret. Ontology querying typically employs query languages designed for ontological data, such as SPARQL [47] for RDF-based ontologies. The goal of ontology querying is to extract relevant and precise information to answer specific questions. Queries may involve searching for instances of classes, relationships between entities, or exploring the hierarchy and properties defined in the ontology.

1.7/ CONTRIBUTIONS

In this thesis, we propose new approaches to enhance the performance of DNNs by incorporating domain-specific knowledge for autonomous driving applications. When we talk about “performance", we are considering a range of improvements. This includes not only getting better computer vision task results based on evaluation metrics but also speeding up the training process, reducing the need for extensive resources, and making more efficient use of raw data. This collectively contributes to more efficient and effective DL models. Our research work involves addressing three fundamental questions:

- What? The first question involves defining what constitutes meaningful knowledge for a specific computer vision task. This includes identifying the domain-specific knowledge that can significantly improve the task performance.
- How? The second question is about how we represent and combine this knowledge with the DL process. We explore various ways to present this information, such as using semantic knowledge, ontologies, or knowledge bases.
- Where? The last question tackles where in the DL process we should incorporate this knowledge to get the best results. Should it be part of the initial data, into the model architecture, or introduced during the models training? Finding the right integration points is the key to optimizing the performance of DNNs.

To address these research questions effectively, one approach is to define specific computer vision tasks and develop frameworks designed to not only outperform the current task performance in the State Of The Art (SOTA) but also enhance the overall DNNs performances. Consequently, our thesis focuses on two major and novel tasks within the field of computer vision: Monocular Depth Estimation (MDE) and panoptic segmentation in urban environments. These tasks are of significant relevance, as they are essential for the development of robust autonomous driving systems. Depth estimation plays a critical role in understanding the three-dimensional aspects of the urban environment, which is essential for safe navigation. On the other hand, panoptic segmentation offers a holistic view of the surroundings by identifying object instances and their semantic categories, a crucial aspect of decision-making in complex urban scenarios. Although these two tasks have some similarities, they also have distinct characteristics that highlight the requirement for different knowledge systems to enhance their performance effectively. Our research, by centering on these tasks, aims to offer valuable insights into how integrating knowledge can improve the performance of DNNs in addressing a wide range of computer vision challenges. In doing so, we contribute to the advancement of more robust and reliable autonomous driving systems.

1.8/ Outline of the PhD thesis dissertation

To offer a comprehensive overview of the contributions outlined in this thesis, the remainder of this manuscript is structured as follows.

- In Chapter 2, we review knowledge-based approaches for computer vision tasks. In the first part, we discuss the use of ontologies as knowledge bases to combine with DNNs, either to verify the results of the networks or to directly enhance the training process. In the second part, we explore other state-of-the-art approaches that combine various forms of knowledge, aside from ontologies, with DNNs.
- In chapter 3, we introduce a new approach that uses an ontology model in a DL context to represent the urban environment as a structured set of concepts linked with semantic relationships. Monocular cues information is extracted through reasoning performed on the proposed ontology and is fed together with the RGB image in a multi-stream way into the DNN for monocular depth estimation.
- In chapter 4, we introduce a new informed DL approach that combines the strengths of DNNs for panoptic segmentation with additional knowledge about spatial relationships between objects. The proposed approach involves introducing a process for extracting and representing spatial relationship knowledge, which is incorporated into the training using a specially designed loss function.

- In chapter 5, we present a general conclusion including the summary of the thesis, and discuss perspectives and future work.

2

STATE OF THE ART

2.1/ INTRODUCTION

In the AI field, significant attention has been given to the fusion of KBS with DNNs, giving rise to the concept of hybrid AI. This chapter provides an overview of the latest approaches that use the power of hybrid AI by combining KBS with DL approaches to enhance the accuracy of different tasks. This hybrid approach represents a fundamental transition in AI research, bridging the gap between structured knowledge and data-driven insights. In the SOTA, two predominant approaches have become well known in hybrid AI: the use of knowledge to improve DL models [48; 49; 50; 51] and the use of DL models to enrich KBS [52; 53; 54; 55]. In the context of our thesis, we will mainly focus on the first category where KBS plays a crucial role in enhancing DL model performance.

Within this category of hybrid AI, we come across various KBS, each with its unique representation and different ways of integrating and combining this knowledge with DNNs. We will categorize the state-of-the-art approaches into two main categories: Level 1 (L1) and Level 2 (L2) approaches. In L1 methods, KBS serve as tools and referees for the validation and the verification of the DL results rather than directly enhancing the training process of the models. In this case, the KBS act as post-processing mechanisms, ensuring the constancy and interpretability of DL outputs. On the other hand, we will explore L2 approaches, where the knowledge is deeply integrated into the training process of the DL models. In this subcategory, KBS plays a more active role in improving the model performance. These approaches frequently involve knowledge at different stages of the training process, significantly impacting the DL model from different perspectives. While we will provide a brief overview of L1 approaches, our primary focus will be on L2 methods. We are particularly interested in exploring approaches where KBS and DNNs are seamlessly integrated into the entire pipeline, working in collaboration to enhance the model performances.

Within this diverse landscape, we propose in the aim of this thesis a categorization of

L2 approaches based on the integration level in the DNNs. These levels include various dimensions, such as integration in early training stages, in the DNNs architecture, or in the last stage.

- **Early stage:** In this category, we present approaches that use knowledge extracted from KBS during the early stages of the DNN training process. This may involve incorporating knowledge into the input of the model through additional data or integrating additional information to enhance the initial training stages.
- **Architecture:** In this category, we explore approaches where KBS are integrated into the architecture of DNNs as an integral component or block. This level of integration enables direct interaction between the model and the knowledge system, improving its performance during the training process.
- **Last stage:** This category includes techniques where KBS are not limited to conventional input or architectural modifications but have a role in the last stage, directly in the prediction of DNNs, by proposing some new loss functions.

2.2/ L1 APPROACHES: KBS FOR DNNS RESULTS VALIDATION

In this section, we present some approaches that make use of KBS to verify and validate the results produced by DNNs. Instead of directly helping with the DNNs training, KBS act like trusted referees, ensuring that the DNNs outcomes are not only accurate but also understandable and explainable. This attention to interpretability is crucial because it allows one to gain insights into how AI systems take decisions, making them more transparent and reliable.

2.2.1/ OVERVIEW OF THE L1 APPROACHES

The approach presented in [28] automated the process of reasoning about errors that emerge from ML algorithms. The goal is to provide explanations for ML errors by using spatial and geometrical reasoning between objects in a scene. The system is demonstrated in the remote sensing domain [56], specifically in pixel-wise semantic segmentation of objects in satellite images [57]. Misclassification is a real challenge in this domain due to visual similarities between classes. For instance, urban infrastructures like buildings or roads often share similar characteristics, contributing to potential misclassifications. The methodology proposed in this paper focuses on spatially explaining the errors in terms of their structure and neighborhood, aiming to improve understanding and interpretation of the learning process. The approach outlines the use of Convolutional Autoencoders (CAE) [58] as the primary ML classifier. Furthermore, the system integrates

ontology-based reasoning using OntoCity [59], which plays an important role in error explanation. OntoCity improves the analysis by decoding the semantics and relationships within the satellite image data, enhancing the capability of the proposed framework to explain errors. While the approach in this paper mainly focuses on explaining ML errors using ontologies, the authors propose some interesting future work. Subsequent research includes using these explanations to not only understand but also improve algorithms performances.

Another approach [27] proposed a method for the classification of healthy fundamental tissues and organs in histological images using an existing histological ontology of the human cardiovascular system [60]. In this paper, two main strategies are presented. The first strategy involves defining discriminant classes that correspond to tissues associated with specific organs, e.g., cardiac muscle, the smooth muscle of arteries, and non-discriminant classes that include tissues that are not directly linked to organs. Resource Description Framework (RDF) triples [41] are constructed based on this ontology, and a reasoner, such as Pellet [36] or FaCT++ [39], is used to perform inference. If the inference outcome is empty, indicating uncertainty, the relevant image blocks are subject to reclassification based on the behavior of false positives. The second strategy addresses the recognition of epithelial tissue through ontology. This category of tissues is typically identified based on the presence of light regions in histological images. The proposed approach extends this recognition to images captured with a $10\times$ objective by considering factors such as the size of light regions and their proximity to specific muscle tissues. RDF rules are generated, and SPARQL queries [47] are used with the ontology reasoner to define the presence of epithelial tissues. Possible outcomes include identifying the type of existing epithelial tissue or defining a high probability of its absence. The paper reports results on F-scores [61] for both organ classification and epithelial tissue recognition. These F-scores are used to evaluate the performance of the proposed methods in classifying various tissue classes. Finally, the paper outlines potential future research directions, including the exploration of CNN [62] strategies and the application of ontology-based classification to various medical problems.

Another paper introduces an approach to explain the behavior of trained artificial neural networks by using semantic web technologies and ontologies [63]. The method aims to provide human-understandable explanations for the network input-output behavior. The authors use description logic and the DL-learner tool to generate explanations based on background knowledge obtained from structured data available on the web. Using the Suggested Upper Merged Ontology (SUMO) [64] as the symbolic knowledge model, the DL-Learner demonstrates the capacity to classify the images through the process of reasoning about the objects based on the ontology defined concepts. Similarly, the methodology introduced in [65] leverages a general purpose ontology, ConceptNet [66], for image retrieval task. The ontology plays an important role in the scoring and ranking

of images during retrieval. When a query includes words that lack pre-trained object detectors, ConceptNet is employed to estimate the likelihood of these words appearing in images. The ontology knowledge is used to enrich the scoring process, allowing for more informed ranking and retrieval of images.

The approach presented in [67] highlights a method for integrating color knowledge into DNNs for indoor object recognition. The process begins by merging two datasets: the public indoor dataset and a private dataset consisting of Frames from Videos (FoVs). This combined dataset is used to train a CNN for object recognition. To incorporate color knowledge, mean images are generated for each object class within the indoor dataset. These mean images serve as representations of typical colors associated with each class. When the network receives a detection request during inference, it uses color knowledge by computing the distance between the input image and the mean images of all classes. This calculation produces a class weight vector, which essentially quantifies how close the input images match the typical colors of different object classes.

2.2.2/ Discussion and analysis

The application of KBS to validate and verify DNNs has gained significant attention in recent research. This type of approach is promising, especially in terms of providing explanations for the perception and decision-making process of DNNs systems. By using KBS, researchers have shed light on the black-box nature of DNNs, making their outcomes more transparent. We have also noticed that the majority of studies in this domain have indeed chosen ontologies over other KBS representations for this category of approaches.

Other KBS representations often fall short in comparison to ontologies when used for the validation of DNNs. One key reason is their formal and structured nature, which provides a well-defined framework for representing complex domain knowledge. Unlike other KBS representations that may lack standardized relationships and semantic consistency, ontologies offer more effective methods for organizing and representing knowledge. Their ability to capture rich, interconnected domain information with clearly defined taxonomies and logical inferences makes them especially effective for the complex reasoning and explanation required in DNNs validation. Moreover, ontologies are more flexible to use in conjunction with semantic web technologies, making it easier to access and query large knowledge bases.

However, while the use of KBS to verify DNNs is a promising initial step, the true potential lies in moving beyond explanation and validation toward collaboration. Integrating knowledge directly into the main pipeline of DNNs allows active collaboration and better results. This integration enables DNNs to use domain-specific knowledge during the perception

process, leading to more accurate and context-aware predictions. In the next section (Section 2.3), we will delve deeper into state-of-the-art methods that integrate KBS into the main pipeline of DNNs to improve their results and performance.

2.3/ L2 APPROACHES: KBS TO IMPROVE DNNS PERFORMANCES

In this section, we present state-of-the-art approaches that use KBS to enhance the performance of DNNs. To provide a comprehensive overview, we categorize these approaches into three distinct subsections, each representing a different integration stage of KBS into DNNs (presented in the introduction of this section).

2.3.1/ EARLY STAGE INTEGRATION

An approach to improve the accuracy of video tagging is introduced in [68]. The method is based on the integration of ontology knowledge into CNN. The key innovation lies in incorporating a video scene ontology, which serves as a structured knowledge representation of the relationships between different classes and concepts within video content. Integration of the ontology occurs at both the input and output layers of the network. In the input stage, each keyframe is labeled with specific ontology classes that represent the content or context of the frame. This labeling is performed through a process known as one-hot encoding [69], where each class is represented as a binary vector. In this vector, 1 corresponds to the class, and 0 is placed at all other positions. These one-hot encoded class vectors are then concatenated with the feature representations of the keyframes. This process reinforces the input data with semantic context extracted from the ontology.

In the same context, another approach is proposed to improve semantic segmentation task through ontology knowledge integration [70]. In this paper, the authors propose a Collaborative Boosting Framework (CBF) that combines data-driven deep learning with knowledge-guided ontological reasoning to improve semantic segmentation in Remote Sensing imagery (RS). The core of this approach lies in the use of ontology knowledge, particularly Remote Sensing Ontology (RSOntology), which serves as a formal representation of domain-specific knowledge. This ontology describes the attributes of objects in the images and the relationships between them, providing a structured foundation for knowledge integration. The DL model used in the CBF is based on a U-Net structure [71]. On the other hand, ontology knowledge is integrated into the DNN in a two-step process. First, intra-taxonomoy ontology reasoning is applied. This involves the direct correction of misclassification in the DNN output based on ontology reasoning rules. These rules are designed to address inconsistencies and improve the accuracy of the initial classification

results. Second, extra-taxonomoy ontology reasoning is used. In this step, additional information such as shadow and elevation is extracted, guided by ontological reasoning rules. This information represented as inferred channels, in addition to the raw image, is integrated as input into the DNN for further iterations that improve both the interpretability and classification accuracy of the DL model.

Other categories of papers have recognized the potential of KBS in enhancing the performance of DNNs in the initial training stages by integrating data extracted from KBS to enrich DNNs. An application of this approach in the SOTA is the generation of synthetic data through simulations based on KBS [72]. This synthetic data serves as a valuable resource for training and testing DNNs. An example of this approach can be observed in the field of AVs, where the synthetic data extracted from KBS are used to augment the training dataset, addressing scenarios that are not represented in the available dataset.

A hybrid modeling method was proposed to integrate ML and simulation techniques [73]. The approach highlights the benefit of combining the two components, with ML strength to handle data and simulation expertise to represent relationships. The authors discuss various integration strategies, such as using simulations to augment training data. This strategy involves creating simulation models to mimic real-world systems, generating synthetic data through simulations, and then combining these synthetic data with real-world training data. The augmented dataset is used to train ML models, improving their performance by providing a more diverse and comprehensive set of examples. An example of this strategy was proposed in the SPIGAN framework that addresses the challenge of unsupervised domain adaptation [74], i.e., an ML algorithm where a model trained on a source domain with labeled data is adapted to perform well on a target domain with unlabeled data without direct supervision in the target domain, in the context of computer vision, with a particular focus on the semantic segmentation task [72]. Specifically, SPIGAN introduces an auxiliary Privileged Network (P) designed to predict crucial information, such as depth, from a simulator. This extracted knowledge, which is not available in real-world data, is then used to train the DNN for semantic segmentation. This knowledge, referred to as Privileged Information (PI), effectively solves issues like artifacts and enhances the DNN capacity to generalize and reduce the domain gap between synthetic and real images. Other approaches also follow the same approach of using KBS to generate synthetic data [75; 76]. These approaches take advantage of the Social Force Model [77] (SFM) as a KBS. This model simulates interactions between individuals and is used to generate realistic scenarios allowing for more robust DNN training and hyperparameter optimization processes.

2.3.2/ Integration of knowledge into the DNNs general architecture

In this section, we present state-of-the-art models techniques that seamlessly integrate knowledge from KBS into the structure and architecture of DNNs.

To improve Zero-Shot learning (ZSL) [78], i.e., an ML approach where a model is trained to recognize classes it has never seen during training, an approach based on ontology knowledge is proposed [79]. It is applied to two distinct tasks: Animal Image Classification (AIC) and Visual Question Answering (VQA). For AIC, a dedicated ontology is constructed, including taxonomic relationships between animals and their visual characteristics. This ontology serves as a fundamental resource for semantic understanding. In the case of VQA, a specialized ontology is extracted from the ConceptNet knowledge graph [80] to model the relationships between answer concepts in VQA questions. The integration of ontology knowledge takes place during the training process. Ontology embedding is used to translate logical axioms and textual information from the ontology into vectors that effectively represent the semantics of class labels. The embedding process involves mapping logical axioms into a geometric space, with loss functions calculated based on geometric inclusion and translation operations, using simple axioms from the ontology. These ontology embeddings are subsequently concatenated with other semantic encoding, such as label word vectors or attribute vectors, depending on the specific task requirements. Ultimately, during the prediction phase, the trained model benefits from the integrated ontology knowledge to classify data into both seen and unseen classes, thereby improving its accuracy by establishing a better understanding of class semantics.

The methodology presented in [81] describes an innovative approach to fine-grained visual classification, specifically targeting the identification of different varieties of fruits within the images. The approach uses a designed ontology to represent essential information about fruit varieties, their attributes, and contextual relationships. In this paper, the ontology integration occurs at multiple stages of the DL pipeline. First, during the training of Mask R-CNN [82] for object detection, ontology plays a crucial role in guiding the network. The ontology provides guidance on what constitutes target objects, e.g., fruit varieties, and their contextual objects such as leaves. This knowledge helps the network focus on accurately identifying these objects within images. Second, when extracting visual attributes from the detected objects, the ontology helps to define which attributes are relevant for each specific object type. For example, it guides the system to recognize attributes like “FruitStripes" for fruit varieties and “LeafEdge" for leaves. These attributes are essential for fine-grained classification. Finally, the structured knowledge from the ontology is integrated into a belief propagation network. This network uses the probabilistic relationships defined by the ontology to refine the classification process. The ontology role here is to provide a structured, domain-specific context that informs the relationships

between objects and their attributes. This context is used to compute the initial evidence in the Bayesian graph [83] and propagate belief for an accurate classification. Integrating the ontology into the entire DL pipeline significantly enhances the model ability to perform fine-grained visual classification by providing critical guidance to object detection, attribute extraction, and belief propagation.

Another paper proposed an algorithm that effectively merges DNNs with first-order logic rules, i.e., they provide a flexible declarative language for communicating high-level cognition and expressing structured knowledge, to enhance the performance of DNNs in various applications [84]. The core of this algorithm includes training the DNNs using labeled data while seamlessly incorporating logical rules to capture some structured knowledge and intentions. The method is based on a unique iterative distillation process [85] that gradually transfers rules-based insights into the DNNs parameters. This transfer process is carried out through the construction of a teacher network that includes the concept of posterior regularization [86]. Namely, the algorithm offers the capacity to find a balance between two important aspects of training DNNs. First, during the early stages of training when the student network is not trained and produces low-quality predictions, the algorithm leans more towards imitation. In other words, it pushes the student networks to mimic the predictions made by the teacher network, which is constructed based on predefined rules. Second, as training progresses and the student network improves its predictive abilities, the algorithm allows the student network to gradually shift its focus toward emulating the teacher predictions. This means that as the student network becomes more trained, it relies less on imitation and more on closely matching the predictions made by the teacher network. This adaptability in balancing imitation and emulation is a crucial feature of the algorithm that ensures effective knowledge transfer. The paper demonstrates the effectiveness of the proposed algorithm on sentiment analysis and entity recognition tasks, showing significant improvement over the base networks.

A paper introduced an approach that uses the SFM [77] as a fundamental knowledge base to improve human motion prediction within a DNN framework [87]. In this fusion of physics-based knowledge and DNNs, the SFM, which includes the fundamental dynamics that control human motion, is integrated into the DNN during the training process. Integration involves incorporating the SFM equations directly into the DNN structure. Within the network architecture, distinct branches are designed to handle various scenarios, such as open and structured environments. Throughout the training phase, the DNN learns how to align its predictions with the knowledge encoded within the SFM. This combination helps the DNN to predict human motion while following the SFM concepts, which leads to better predictions and adaptability to various scenarios beyond the training data. Other approaches have been proposed in the same context of human motion or pedestrian trajectory prediction, leveraging physics-based models, mainly SFM, in combination with DNNs during training [88; 89]. The integration of physics knowledge is important for

such tasks because it provides a deep understanding of the dynamics and interactions in the real world. Physics-based models describe how individuals or agents move based on physics fundamentals, such as forces of attraction and repulsion. By incorporating these insights into DNNs during training, the models gain a more comprehensive understanding of human behavior, leading to more accurate predictions, especially in complex scenarios.

Many researchers are focusing on Gated Graph Neural Networks (GGNNs) [90], a type of DNNs based on earlier Gated Neural Networks (GNNs) [91]. The goal is to include contextual knowledge during model training through the network structure. GNNs are designed to handle information organized in a graph where each node corresponds to a hidden state vector that is updated iteratively. GGNNs allow information to move in both directions (forward and backward) and can update multiple nodes simultaneously at each step. In this context, an approach was proposed to use contextual information, including object types and spatial relationships [92]. This approach aimed to detect action-object affordances effectively. Furthermore, another method proposed a two-step process that employs a CNN to identify functional areas within indoor scenes [93]. More recently, a GGNN was used to consider the overall context of a scene for object detection. This approach also suggested the most suitable object for a specific task [94]. Finally, the approach proposed in [95] focused on situation recognition tasks. The goal is to identify human-object interactions within a given context by predicting the most suitable verb that describes the ongoing activity in a scene. The authors used a GGNN, that integrates reasoning about verbs and their corresponding roles by iteratively transmitting messages along the graph edges.

The approach proposed in [96] introduced a framework that combines DL with additional knowledge represented as a knowledge graph to improve visual recognition. The framework consists of two main modules: a local module that uses spatial memory and a global graph-reasoning module. The knowledge graph encodes semantic relationships between object classes, providing valuable structured information. Edge weights in the graph capture relationships and influence information flow. The model performs message propagation on the graph for reasoning, allowing it to consider both local and global contexts. Cross-feed connections ensure collaborations between local and global modules. During training, the model learns to use the knowledge graph. This combination of spatial and semantic reasoning, along with attention mechanisms, improves recognition performance.

An approach to enhance DL models for image classification of real-world event types is presented in [97]. This approach is based on the integration of structured knowledge extracted from an ontology into the DL model. The ontology is constructed using a large knowledge base, providing a rich taxonomy of event types. The DL model used

in this study is based on the ResNet-50 architecture [31], fine-tuned for event classification tasks. In the proposed method, DNN processed the input images, while the ontology improves the model understanding of event semantics. This fusion involves two main components: a classification approach and an ontology-driven network. The classification approach predicts a leaf node vector, providing probabilities for a subset of event nodes that can be used directly for classification. The ontology-driven network, on the other hand, outputs a sub-graph vector with probabilities for all event nodes in the ontology. The specific mechanisms of the ontology integration into DNNs involve multiple strategies, such as measurements of cosine similarity [98] and elements-wise products, to combine the output of the classification approach and the ontology-driven network.

A multi-modal framework for autonomous robotics environment representation is presented in [51], aiming to bridge the gap between data-driven methods and semantic understanding. The approach consists of three main units: perception, instance, and knowledge. In the perception unit, a modified AlexNet / VGG model [99] processes raw sensor data for scene segmentation, object detection, and instance tracking. The knowledge unit constructs an ontology using WordNet [100] and syntactic analysis. It represents concepts and relations about objects descriptions and functions. Finally, the instance unit links real-world observations to semantic concepts and serves as a bridge between sensor data and knowledge. User requests destined for the robot are converted into dynamic Prolog predicates [101], initiating tasks, and reasoning between spatial relations between instances. The system combines ontology-based knowledge with DNNs during task execution, improving the robot ability to understand and interact with its environment.

The methodology outlined in [102] presents an Ontology-Based Semantic Image Segmentation (OBSIS) approach that aims to improve image segmentation by bridging the gap between low-level visual features and high-level semantic knowledge. OBSIS employs a combination of techniques, including a DL model, semantic ontology, and probabilistic graph model. In OBSIS, a DL model, specifically a CNN, is used to analyze and extract low-level features from images. A semantic ontology is constructed using the OWL 2 DL language [103], representing high-level semantic knowledge and relationships between objects, their parts, and visual features. The knowledge integration step occurs in the intermediate semantic space. After extracting low-level features, Dirichlet process mixture models [104] and Conditional Random Fields (CRFs) [105] are used to transform the visual features into this higher-level space. In this space, the features are represented as intermediate labels associated with color, texture, and shape. The ontology is then used for the final inference, enabling the extraction of semantic labels by capturing interactions between semantic concepts and visual features in a semantic context model.

2.3.3/ Integration of knowledge in the last stage

A widely employed method for enhancing the outcomes of DNNs involves the use of knowledge-guided loss functions. These loss functions are designed to ensure that the model outputs align with knowledge bases and rules, eliminating unrealistic predictions that are incompatible with real-world scenarios. The approach proposed in [106], provides a methodology for constructing physics-guided DL models. It identifies the expanding interest in this field, with diverse applications in many domains. The authors explore the design of loss functions that incorporate physics-based constraints. These loss functions guide the training of DNNs to produce results that are not only data-driven but also physically consistent.

Another approach introduced in [107] focuses on improving the performance of DNNs for constrained problems, specifically the Partial Latin Square (PLS) completion problem. The main goal is to improve the ability of DNNs to general practical solutions by integrating domain knowledge into the training process. This integration is achieved through a novel loss function that goes beyond typical data-driven terms. The loss function incorporates both data-driven terms, such as cross-entropy, and additional terms inspired by Semantic Based Regularization (SBR)[108], i.e., an ML technique that incorporates semantic information such as class relationships or embeddings, as a regularization term during training to enhance model generalization, and Constraint Programming (CP) [109], i.e., a declarative programming paradigm focused on expressing and solving complex problems through constraints and variables. These additional terms penalize the model based on domain-specific constraints, guiding the network to produce solutions that are not only data-driven.

A widely acknowledged challenge associated with Generative Adversarial Networks (GANs) is their sample complexity, which demands a large amount of data for effective training. GANs are type of ML models that include two neural networks, a generator, and a discriminator, trained simultaneously through adversarial training. The generator aims to create realistic data, while the discriminator tries to distinguish between real and false data, leading to a dynamic learning process that enhances the generator ability to produce increasingly realistic outputs. There is a research area dedicated to enhancing GANs by incorporating prior knowledge of physics, using physical laws and invariance properties. For example, in the context of predicting turbulent flows, GAN-based models have shown improved performance when integrating physical constraints, such as adhering to conversation laws [110] and the energy spectrum [111] into the loss function. In this context, the work proposed in [112] applied a physics-based morphology constraint to a VAE-based GAN model to simulate artificial material samples. In the same context, many approaches have shown great success in enhancing DNNs through the integration of physics-based loss functions. For example, in lake temperature modeling,

a proposed approach added a physics-based penalty term to make sure that the predictions followed a straightforward pattern: denser water should be predicted to be at lower depths compared to less dense water [113]. Other works went a step further in this research direction by using more intricate physical relationships [114; 115]. They introduced a physics-based constraint in the loss function to maintain the balance of thermal energy in the lake over time, aligning it with the net thermodynamic exchanges between the lake and its surroundings.

Another approach introduced a hybrid modeling methodology that integrates domain-specific knowledge into DNNs by incorporating it directly into the loss function [116]. This approach aims to enhance the accuracy and generalization capability of DNNs, particularly when dealing with sparse and noisy process data. The hybrid model leverages prior knowledge through the inclusion of simple process models and first-principles equations, effectively integrating this knowledge as constraints within the global loss function. These constraints guide network predictions in regions of the input space with limited training data.

2.3.4/ Discussion and analysis

In Section 2.3, we have explored the integration of KBS into DNNs from three distinct stages: early stage, during training, and last stage integration. We also presented a summary of the described approaches in Table 2.1. Each of the three integration methods brings its own set of advantages and limits. Early stage integration approaches involve incorporating structured knowledge directly into DNNs at the input stage. This approach offers a strong foundation for DNNs by enhancing raw data with additional knowledge. One important advantage is the ability and potential to improve predictions since initial iterations. However, a significant challenge in early stage integration is that the effectiveness of the approach relies heavily on the quality and completeness of the integrated knowledge. It should be strong, precise, and adequate for the target task to ensure that the information injected into the DNN is helpful and meaningful. This requires accurate knowledge expertise to keep the knowledge relevant.

During training integration approach is more dynamic since it actively involves KBS in the training process. Rather than just enhancing input data, it integrates knowledge into the architecture of DNNs or guides their learning through specialized loss functions. This approach allows for iterative knowledge injection, enabling DNNs to correct wrong predictions and acquire additional information during training. The adaptability it offers is a significant advantage, as it allows DNNs to continuously refine their understanding. However, it is important to note that this approach can lead to increased computational complexity and longer training time, especially when dealing with complex knowledge

Approach	Task	KBS	DNN
Early stage integration			
[68]	Video tagging	Video scene ontology	CNN
[70]	Semantic segmentation	RS-Ontology	U-Net
[70]	Semantic segmentation	Simulation model	GAN
[75]	Crowd analysis	SFM	CNN
[76]	Human trajectory prediction	SFM	RNN
Integration into the DNNs architecture			
[79]	Image classification	Ontology	RNN
[81]	Image classification	Ontology	Mask R-CNN
[84]	Sentiment analysis	Logical rules	CNN
[84]	Entity recognition	Logical rules	RNN
[87]	Human motion	SFM	CNN
[88]	Pedestrian trajectory	SFM	Feed-forward neural network
[89]	Human motion	SFM	auto-encoder
[92]	Affordance detection	Contextual information	Gated-graph neural network
[93]	Scene understanding	Scene functionality ontology	CNN
[94]	Object detection	Contextual information	Gated Graph neural network
[94]	Situation recognition	Logical rules	Gated-graph neural network
[96]	Visual recognition	Knowledge graph	CNN
[97]	Image classification	Event types ontology	RNN
[51]	Scene understanding	Ontology	CNN
[117]	Object detection	Ontology	YOLO
[102]	Image segmentation	Ontology	CNN
Last stage integration			
[112]	Turbulent flow prediction	Physics constraints	GAN
[107]	Constraint problem	Domain knowledge	CNN
[113; 114; 115]	Environmental modeling	Physics constraints	CNN
[116]	Environmental modeling	Simple process models	Radial basis function neural network

Table 2.1: Summary of Level 2 state-of-the-art approaches.

structures. Last stage integration provides flexibility by introducing KBS into the final stages of DNN training, directly influencing the model output. This approach is advantageous for integrating domain-specific knowledge into existing DNN architectures without requiring major architectural changes. However, it may not provide the same level of interpretability and knowledge fusion as the first two approaches.

In summary, the choice of when and how to integrate knowledge into DNNs depends on the specific task, resource availability, and the desired trade-offs between interpretability, accuracy, and computational complexity. KBS integration has shown significant promise in enhancing DNN performance, but ongoing research is needed to address challenges such as knowledge representation, particularly in dynamic and complex domains like AV.

2.4/ CONCLUSION

In conclusion, this state-of-the-art chapter provides a comprehensive overview of the latest approaches in the field of hybrid AI, where KBS are seamlessly integrated with DNNs to enhance the accuracy and interoperability of AI systems. Hybrid AI represents a significant transition in AI research, bridging the gap between structured knowledge and data-driven insights. The chapter classified hybrid AI approaches into two main categories: L1 and L2 approaches. In L1 methods, KBS primarily serve as tools for validation and verification of DNNs results, ensuring the reliability and interoperability of DL outputs. On the other hand, L2 approaches involve deep integration of knowledge into the DNNs training process, significantly impacting model performance.

Within L1 approaches (Section 2.2), we explored various strategies, particularly the use of ontologies and other structured knowledge, to explain and validate DNNs results. These approaches have shed light on the interpretability of AI systems and their decision-making process. Ontologies have proven to be effective in organizing and representing complex domain knowledge for improved validation. Moving beyond L1, L2 approaches (Section 2.3) directly integrate the knowledge from KBS into DNNs at different stages of the training process. Early stage integration enhances training data with additional information extracted from knowledge structures, improving the training from its initial stages. Integration during the training process actively involves KBS in the training pipeline, enabling DNNs to leverage domain-specific knowledge. Last stage integration allows KBS to influence the final output of DNN, which impacts the overall performance of the model.

Our primary focus in this exploration of hybrid AI has been oriented toward L2 approaches, primarily due to the belief that these methods represent a more advanced concept to seamlessly combine KBS with DNN. L2 approaches demonstrate a deeper integration of structured knowledge through the entire DNN training pipeline, allowing active

collaboration between KBS and DNNs to enhance network performance. An important observation is that these approaches initially found widespread use in fields like physics, environmental sciences, and other scientific domains where physics-based knowledge is helpful. However, over time, these approaches have also shown effectiveness in computer vision, where semantic understanding and context-aware predictions are essential. Nevertheless, it is interesting to note that these approaches are relatively less common in the AV field. This can be attributed to the complexity and dynamic nature of the autonomous driving environment, making it challenging to define and integrate the knowledge required effectively.

In the context of this thesis, our research will align with the general methodology of L2 approaches, particularly when applied to the challenging domain of AV. Our primary goal is to propose innovative approaches combining KBS with DNNs to enhance the perception capability of AVs operating in dynamic outdoor environments. The challenge lies in defining and acquiring knowledge that holds meaningful relevance for computer vision tasks in these complex outdoor scenarios. This environment is characterized by many factors, including changing weather conditions, unpredictable traffic patterns, and diverse road structures. Within this general framework, we will explore multiple strategies for integrating knowledge into DNNs, with a particular focus on incorporating knowledge both statically at the input stage and dynamically during the training process.

3

INTEGRATION OF ONTOLOGY REASONING-BASED MONOCULAR CUES IN DEEP LEARNING MODELING FOR SINGLE IMAGE DEPTH ESTIMATION IN URBAN DRIVING SCENARIOS

3.1/ INTRODUCTION & CONTEXT

In this chapter, we present our first contribution within the scope of this thesis. We propose a new hybrid approach that combines KBS with DNNs to improve monocular depth estimation task in the context of urban driving. In this section, we introduce the general context of this work, with a focus on depth estimation task. The existing categories of methods, their strengths, and limitations are presented. Afterward, we delve into the motivations and intuitions behind the proposed approach.

Depth estimation has long been acknowledged as a fundamental task in the computer vision field, improving the ability to perceive and understand scenes in various applications, including autonomous driving [118]. However, the process of generating accurate depth maps is usually expensive and requires considerable computing resources. Consequently, depth estimation based on computer vision techniques has become a focus of interest in the scientific community [119]. In this context, we classify image-based depth estimation techniques into three main categories: traditional, ML-based techniques, and DL ones.

Traditional methods rely mainly on geometric relationships between visual features gathered from multiple viewpoints, as seen in techniques such as stereo vision matching [120] and vanishing points [121]. The effectiveness of these techniques is highly dependent on

the accurate detection, matching, and tracking of features, which in turn depend on the quality of the image sequence. These approaches tend to be complex, less practical, and not well-suitable for real-time applications [118].

The second category that represents ML approaches is based on methods and algorithms that use parametric and non-parametric ML, such as Markov Random Field (MRF) [122] and Pyramid Histogram of Oriented Graph (PHOG) [123], to name just a few. These methods offer some advantages, including the correlation between scene depth and texture cues obtained from computed texture features at various scales [124]. However, these methods are often criticized for their high complexity, dependence on texture information, and the challenge of real-time execution.

Finally, the most recent advancement in this field can be seen in the third category of DL-based techniques. These approaches, whether supervised, non-supervised, or semi-supervised, stand out for their practicality, accuracy, and efficiency, especially in multi-scene and real-time applications [118]. Within the context of DL, we identify two main categories of depth estimation models: binocular-based and monocular-based.

When it comes to binocular depth estimation models, many researchers have turned to stereo vision to achieve accurate depth estimation results [125; 126]. This approach is based on a binocular rig that allows one to acquire a pair of images to estimate the disparity by stereo matching [127]. However, it is worth noting that this approach requires at least one stereo camera. It is also difficult to capture enough features in the image to generate dense depth maps when the scene has few or no textures [128]. In this context, some researchers have suggested alternative methods to estimate the depth, motivated by the need to reduce the required hardware resources. According to recent work, pixel-wise depth maps could be generated end-to-end from a single image [129]. Several DNNs, including CNNs [130] and VAE [131]), have demonstrated outstanding performance and effectiveness. These promising results have inspired the community to investigate the process performed by DL models to identify the factors that impact MDE.

According to [132; 133], DNNs have demonstrated their ability to learn visual depth cues or any information sent from a two-dimensional image that provides a three-dimensional impression to the observer. These cues, also known as monocular cues, are used to evaluate depth from a single image. Examples of these cues include texture gradients and the apparent size of objects. They represent how a single eye allows us to see and process what we perceive in our environment. For humans, these monocular cues are based not only on the visual aspect of the objects but also on previously learned and acquired knowledge. For example, the following reasoning is automatically performed if a driver perceives a car to be larger than a truck in an urban environment:

1. Perception: The car appears to be larger than the truck.

2. Knowledge: The absolute size of the car is smaller than the absolute size of the truck.

3. Reasoning: According to the perception and the pre-acquired knowledge, we conclude that the car is closer than the truck.

This reasoning is based on the apparent size of the objects, one of the monocular cues, and the knowledge regarding the absolute size of those objects.

According to the SOTA (Section 3.2), many works have shown that DNNs can learn more effectively when they are trained with both semantic and visual information, which helps in depth estimation tasks. Also, it has been shown that DNNs can implicitly learn visual depth cues during their training process. In light of these findings and the above observations, we were inspired in the context of this work to explore the explicit and direct integration of knowledge during the training of DNNs for MDE. We hypothesize that by incorporating meaningful knowledge directly and explicitly during DNN training, we can further enhance the model ability to understand and learn depth cues, leading to improved performance for the considered computer vision task. Consequently, we propose a new DL approach that directly incorporates monocular cues representations as additional inputs into the MDE process. Specifically, we use semantic segmentation to identify objects in the urban scene and extract the relevant knowledge for each pixel in the image. This basic human-like knowledge that leads to monocular cues is obtained thanks to an ontology and rules that use different geometric and spatial information related to the urban environment. These are then fed with the RGB image into a DL model for depth estimation.

The contributions described in this chapter are as follows.

- the definition and implementation of an ontology and a rule base for monocular cues extraction,
- the implementation of the proposed monocular cues on two deep neural networks for monocular depth estimation,
- the validation and evaluation of our approach on various urban scene datasets, including unseen scenes.

To present our approach, the remainder of this chapter is organized as follows. Work related to MDE is introduced in Section 3.2. Section 3.3 outlines the proposed approach. The methodology including ontology reasoning and monocular cues extraction is described in Section 3.4. The DNNs considered for MDE are presented in Section 3.5. Section 3.6 presents the performed experiments, the comparison of the results with the SOTA, and the ablation study. Finally, the last section concludes the work in this chapter and provides directions for future work.

3.2/ MONOCULAR DEPTH ESTIMATION STATE-OF-THE-ART

Many approaches have been introduced to tackle monocular image-based depth estimation. In [134], the authors improved the performance of the MDE DNNs through a straightforward but well-designed model for this specific task. The technique includes the use of minimal re-projection loss to address occlusion, and the incorporation of full-resolution multiscale sampling to reduce visual artifacts. Another method for single image depth estimation without the need for ground truth data was introduced in [135]. This method relies on epipolar geometry constraints to generate disparity maps through network training with an image reconstruction loss. Among the various works related to monocular image depth estimation, the method proposed in [136] is currently considered the SOTA in supervised MDE, to the best of our knowledge. In this work, the authors introduced a novel transformer-based architecture block known as AdaBins. It adaptively divides the depth range into bins to estimate the center value for each image. Ultimately, a linear combination of these bin centers is used to estimate the final depth values.

Since the quality of depth estimation from a single image has rapidly increased, some works [132; 133] have analyzed the process performed by DNNs to understand how the depth is estimated. Their goal has been to identify the factors that DNNs implicitly take into account during the learning process, enhancing their efficiency and precision in depth estimation. In [132], a comprehensive analysis of various DNNs was conducted, focusing on the monocular visual cues employed in depth estimation. Their findings revealed that DNNs primarily rely on perceived monocular depth cues, such as relative size, linear perspective, overlap, and elevation. Practical evaluations of the influence of each depth cue demonstrated that DNNs mainly rely on the vertical position in the image space of objects in driving scenarios. Furthermore, another work [133] also considered CNN inference visualization to gain insights into the depth inference process from single images. Applying their method to different depth estimation networks on outdoor scene datasets, the authors made several key observations. First, CNN behavior indicated that it selects image edges based on the geometry of the scene objects; this means that geometric specifications such as size, volume, length, and height are mainly used in the object edges selection in the image. In addition, DNNs take into account information located within specific regions of each individual object to estimate depth accurately.

Acknowledging the importance of leveraging supplementary information to enhance MDE, various works have introduced approaches to take advantage of this idea. To implement this concept, semantic information has been used to identify the visual content within an image by establishing connections between low-level features and the scene content. This integration of semantic information enables additional knowledge transfer to the depth estimation models, helping them to accurately perform their learning. In this context, a first set of papers [137; 138] proposed adding semantic information to address

the challenges caused by dynamic objects. Furthermore, another work attempted to integrate both semantic segmentation and depth estimation networks into a unified framework. This architecture allows insights issued from the first network (semantic segmentation) to be considered for the enhancement of the subsequent network (depth estimation) [139]. Including semantic information within the process improves depth estimation. This is mainly because semantic information brings additional knowledge to the task workflow.

From a broad perspective, the practice of integrating supplementary data and information to enhance the performance of computer vision tasks has consistently yielded promising results. For example, a novel multi-task DL architecture for face pose estimation was proposed in [140]. This method employs a deep CNN-based feature extraction to represent facial images and a multi-task learning-based model to establish the correlation between images and corresponding poses. The incorporation of multimodal features within the context of multi-task learning further improves performance. Furthermore, the authors in [141] proposed a method for 3D object recognition based on the fusion of multi-view data. This multi-view fusion technique optimizes the performance by computing a refined weighted combination of data, thus enhancing the overall results. Another approach was introduced for 3D human pose recovery [142]. This method relies on the integration of additional information concerning 2D silhouettes. It also introduces the concepts of locality-sensitive constriction and the combination of multi-view features to improve results. Finally, it has also been proposed to improve 3D human pose recovery through the introduction of a Multimodal Deep Autoencoder (MDA) and a nonlinear Backpropagation-Neural Network (BP-NN) [143]. The autoencoder adeptly combines diverse feature types by leveraging multi-view hypergraph Low-Rank Representation (LRR) learning, further enriching the capabilities of the system.

The various approaches discussed in this section shed light on the growing importance of leveraging additional knowledge to enhance the performance of MDE DNNs. The incorporation of extra information stands out as an effective strategy for addressing challenges related to the task and refining depth estimation. Similarly, the analysis of DNNs behavior provided valuable insights into the implicit factors guiding depth estimation. These methodologies, however, have some limitations and challenges. The success and usefulness of integrating additional knowledge depend on the specific task and also the quality of the considered information. Additionally, the computational cost of incorporating this information should also be considered. In light of these findings and considerations, we were motivated to take advantage of the usefulness of contextual knowledge in monocular depth perception and propose a new approach consisting of integrating monocular cues in the learning process of DNNs for MDE. Such an approach may not only enhance the performance of depth estimation but also contribute to a more comprehensive understanding of the context.

In the SOTA chapter of this thesis (Chapter 2), we discovered that one effective way to represent knowledge is through the use of ontologies. They offer many advantages, including structured knowledge representation, semantic clarity, and the ability to capture complex relationships within a specific domain. However, although several papers consider ontologies to enhance DNNs for various computer vision tasks applied in multiple domains, to the best of our knowledge, there is an important gap in the literature concerning the use of ontology knowledge for MDE. Consequently, in this contribution, we aim to propose a methodology that integrates monocular cues knowledge extracted from ontologies into DNNs to improve MDE. This approach has the potential to leverage the structured knowledge represented in ontologies to enhance the performance and contextual understanding of MDE models.

3.3/ Overview of the Proposed Methodology

The general system of the proposed approach is illustrated in Figure 3.1. The system takes as input an RGB image captured within an urban environment. This image is subsequently processed through a pretrained semantic segmentation network, leading to the generation of a semantic segmentation map. The latter is considered to formulate reasoning on the concepts implemented within the proposed ontology. This ontology refers to various geometric, contextual, and semantic information related to the urban road context to represent basic and essential human knowledge.

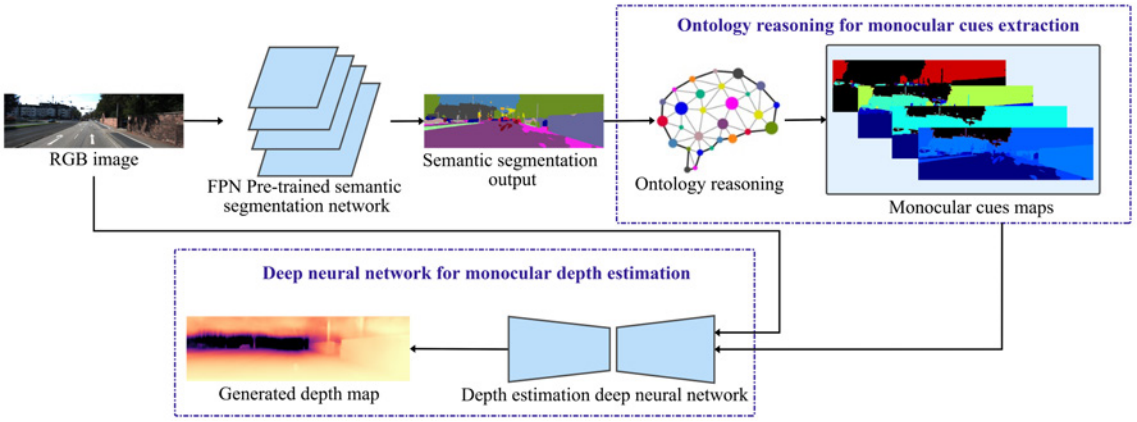


Figure 3.1: General pipeline of the integrated system for monocular depth estimation : Ontology reasoning and deep learning.

Based on this knowledge, ontology reasoning, which imitates human-like reasoning, is performed to extract monocular cues information from the semantic segmentation. Additional insights into the process of ontology creation, ontological reasoning, and monocular cues extraction can be found in Section 3.4. The extracted monocular cues, along with the original RGB image, are seamlessly integrated in a multi-stream way into a DL model

designed for MDE. To validate the proposed approach, we considered two different DNNs, which are detailed in Section 3.5.

3.4/ Ontology reasoning for monocular cues extraction

As part of our contribution, our goal is to emulate human-like reasoning in the context of depth estimation. To achieve this, we have chosen to create an ontology for extracting the required knowledge related to various monocular cues. We have opted for this approach because an ontology is well-suited for defining concepts, properties, and relationships providing a structural and geometrical framework for modeling the urban environment.

3.4.1/ Ontology creation

In this work, we have specifically chosen to employ OWL as the ontological language. OWL is based on descriptive logic and functions that divide knowledge bases into two essential components: the T-Box and the A-Box. The T-Box describes the domain through concepts and relationships, while the A-Box is composed of factual assertions that are interconnected with the conceptual model defined in the T-Box. Within the scope of this work, we have established a T-Box relying on expert knowledge. This T-Box consists of concepts and relationships designed to model the necessary knowledge to enhance depth estimation. This knowledge is general and can be applied regardless of the scene. To further enhance the performance of our ontology and enable more sophisticated rule-based reasoning, we have incorporated Semantic Web Rule Language (SWRL) [144] into our framework. SWRL extends OWN by allowing the creation of rules and adding inference capability to the ontology. These rules facilitate the representation of complex relationships and dependencies, providing a powerful tool for refining our ontology model and supporting challenging applications such as MDE. The creation of an ontology typically involves two key stages: knowledge acquisition and knowledge modeling.

3.4.1.1/ Knowledge acquisition

In the process of acquiring ontological knowledge for a specific use case, the first step is to determine the domain and scope of the ontology [35]. To do so, two essential skills questions are involved: “Which concepts exist in the domain concerned by the ontology develop-ment?" and “Which concepts are the most relevant to the application?". To address these inquiries, we first defined the domain of our ontology, which centers around the urban driv-ing context. Subsequently, we enumerate the important terms in the ontology and identify

concepts related to our specific domain. These concepts are linked to the various components and objects within urban driving scenes. In practice, we characterized the essential contextual, geometric, and semantic knowledge associated with each pertinent object, embedding this information into the ontology to obtain additional relevant indices that will help improve MDE. To ensure the precision and reliability of the knowledge represented within the proposed ontology, we refer to urban environment regulations, extracting the laws governing the construction, layout, and deployment of the different elements in urban environments. We integrate this type of knowledge because the challenges related to MDE are mainly due to the inability to transfer some essential knowledge only from visual information coming from the camera. Hence, the proposed ontology is based on the knowledge that, when processed through ontology reasoning, generates monocular cues (see Section 3.4.2) able to enhance depth estimation.

3.4.1.2/ KNOWLEDGE MODELING

The knowledge modeling step involves the creation of ontology concepts and the establishment of connections between them. This step aims to formalize the conceptual entities identified during the acquisition stage, i.e., to identify the relationships among the ontology concepts and define their properties. The modeling process depends mainly on the needs required by the creation of the ontology. It is important to create generic concepts and a hierarchy to organize the ontology by regrouping concepts with shared characteristics. Typically, these concepts are linked together with a *“hasSubClass"* relationship. For example, the generic concept *“Human"* is linked to more specific concepts such as *“Rider"* and *“Pedestrian"* via the relationship *“hasSubClass"*, creating a hierarchical tree structure with each concept as a child of a parent concept. Moreover, to model the characteristics of these concepts, we define the properties of classes by introducing data properties that connect a single concept to attribute data, and object properties that establish relationships between two concepts. Within the proposed ontology, we have represented the structural aspects of an urban environment to acquire knowledge relevant to the target application domain. Consequently, our choices for data and object properties are made to capture the most generic structural elements and the most typical scenarios found in urban environments, e.g., bikers ride on the roads, roads have sidewalks on both sides, etc. For example, within the proposed ontology, the “Pedestrian" concept (as described in Figure 3.2) is associated with data properties such as volume, height, and distance from the road center, in addition to object properties such as the *“isOn"* relationship that connects the *“Pedestrian"* to the *“SideWalk"* concept.

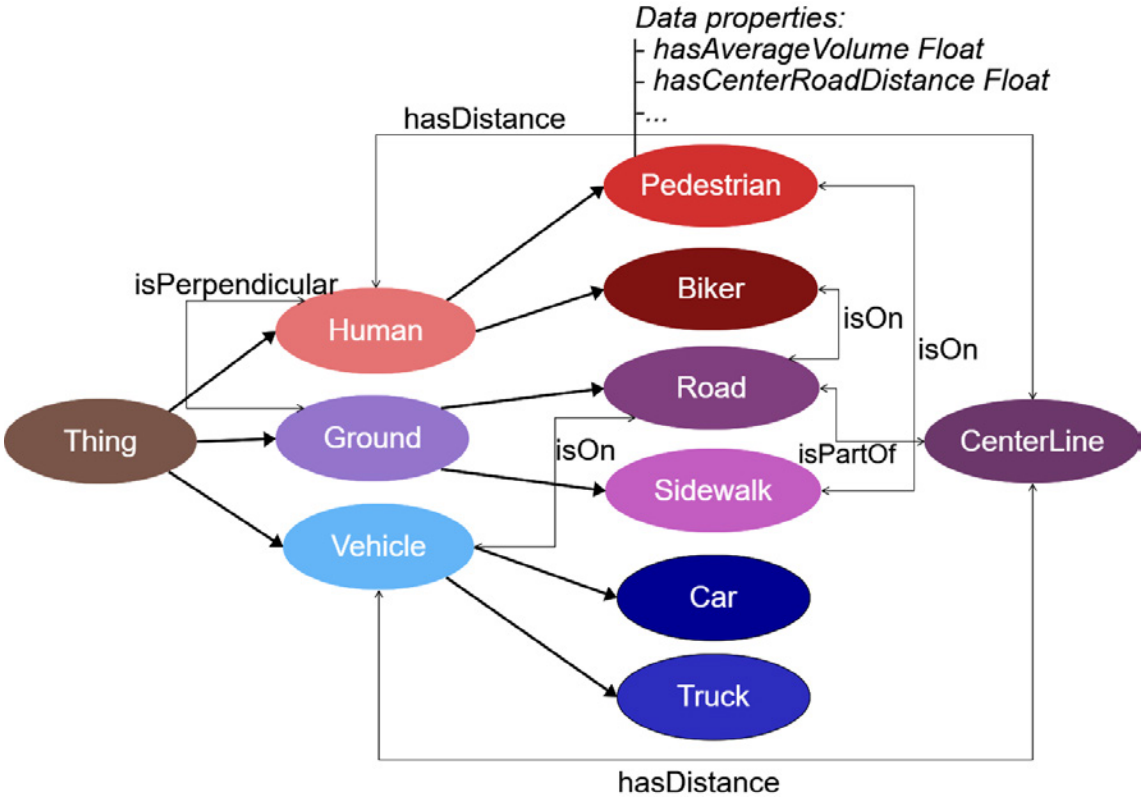


Figure 3.2: Part of the overall ontology and example of data properties on the pedestrian concept.

3.4.2/ MONOCULAR CUES

In the upcoming sections, we outline the general process for extracting monocular cues maps (Section 3.4.2.1). A detailed description of the proposed monocular cues maps and the ontology reasoning process is provided in Section 3.4.2.2.

3.4.2.1/ GENERAL PIPELINE FOR MONOCULAR CUES MAPS EXTRACTION

The proposed ontology is primarily built on open concepts representing objects within the urban environment. In this context, it is crucial to define the target classes specific to the urban environment, allowing the proposed ontology to encapsulate these classes as concepts with their associated knowledge for subsequent ontology reasoning. As illustrated in Figure 3.3, the introduced workflow takes as input an RGB image captured by a vehicle equipped with an onboard camera operating within an urban environment. Considering a state-of-the-art pretrained DNN, we generate a pixel-wise semantic segmentation. While the primary focus of this work is not semantic segmentation itself, we have chosen to use a pretrained state-of-the-art model to assign the relevant class to each pixel in the image. Once the semantic segmentation is obtained, we proceed to create our monocular cues

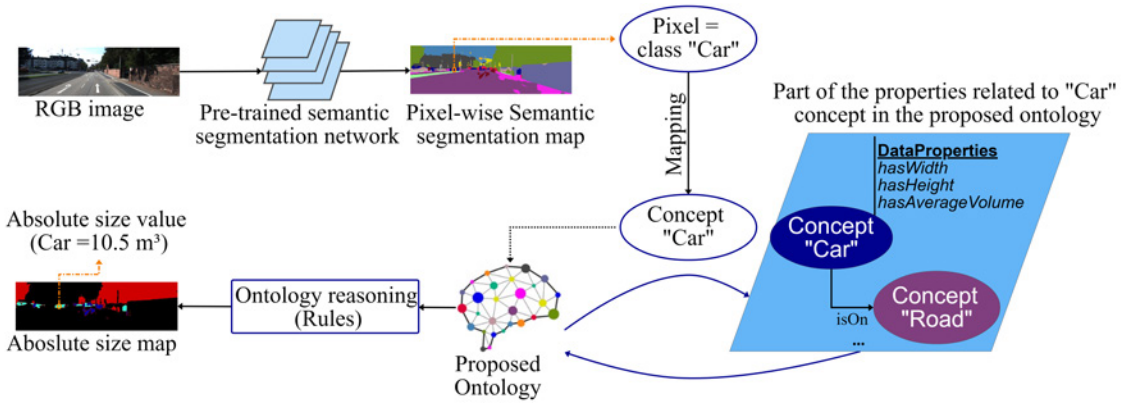


Figure 3.3: Ontology-based process for monocular cues extraction. Example of an absolute size map.

maps.

Let us take the practical example represented in Figure 3.3 to identify the upcoming steps in creating the monocular cues maps. We have considered a specific pixel in the image, located at position (i, j), to illustrate the entire process. In this case, semantic segmentation has classified this pixel as a *"Car"*. This classification directly links the classified pixel with the *"Car"* concept in our previously established ontology. From the ontology point of view, the *"Car"* concept includes various data properties, such as *"hasWidth"*, *"hasHeight"*, and others. Leveraging this knowledge, along with the ontology rules detailed in Section 3.4.2.2, information representing monocular cues could be extracted. In the proposed example, the ontology reasoning indicated that an instance categorized as a *"Car"* concept has an absolute size of $10.5m^3$. Subsequently, we store this information on a map of the same dimensions as the RGB image, precisely at position (i, j) corresponding to the considered pixel. These steps are repeated for all pixels in the image, resulting in the creation of a map that effectively represents the monocular cue "Absolute size" (Further insights into the proposed monocular cues are provided in Section 3.4.2.2)

3.4.2.2/ Description of the proposed monocular cues maps

In the following sections, we explain each of the proposed monocular cues maps, as well as the ontology reasoning used for their extraction.

Distance from the road center The monocular cue referred to as *"Elevation"* is a measure of an object distance in relation to the horizon in the image space. This concept significantly contributes to improved depth perception, as objects closer to the horizon tend to appear more distant, while those located further from the horizon are perceived as closer and more proximate. Taking inspiration from this monocular cue, we have in-

troduced a map that specifically represents the horizontal distance between each object within the urban scene and the center of the road. Figure 3.4c shows an example of this map representation.

Let us begin by examining the straightforward human reasoning applied to estimate the distance of an object from the road center. As an example, consider the estimation of distance, denoted as d, between a *"SideWalk"* and the road center. Human reasoning is based on the geometric knowledge of the environment, specifically the dimensions of the *"Road"* and the *"SideWalk"*. Assuming an average width of the *"Road"* as a and the *"SideWalk"* as b, humans deduce that the *"SideWalk"* is positioned at a distance of $\frac{a+b}{2}$ from the road center. To model this human reasoning process, we have encoded this pre-existing knowledge at the core of the proposed ontology. More precisely, we have incorporated the *"SideWalk"* and the *"Road"* widths as data properties, each assigned defined values that represent the typical average widths of roads and sidewalks within a standard urban environment, as regulated by laws. The reasoning rule within the ontology subsequently calculates the *"SideWalk"* distance from the *"Road center"*. It is defined as follows.

SideWalk(?s) $\wedge$ Road(?r) $\wedge$ hasWidth(?s,?a) $\wedge$ hasWidth(?r,?b) $\wedge$
swrlb:add(?x,?a,?b) $\wedge$ swrlb:divide(?d,?x,2)-> hasCenterRoadDistance(?s,?d)

This rule is structured as an implication between an antecedent (the body or left part of the rule) and a consequent (the head or right part of the rule). Essentially, the rule operated as follows: when the conditions specified in the rule body are met, the conclusions specified in the rule head must also hold true. Within the rule body, we introduced a variable denoted as "$?s$" to represent the concept *"SideWalk"*. This variable serves as a means to browse the ontology and retrieve instances associated with the *"SideWalk"* class. In our specific case, we have defined a generic instance named *"SideWalkx"* to encapsulate the properties of the *"SideWalk"* concept. In the same way, we introduce a variable "$?r$" for the concept *"Road"*. Furthermore, the rule body contains a statement related to the data property *"hasWidth"*. For instance, the statement "$hasWidth(?s, ?a)$" allows the definition of a variable "$?a$" to store the width of the instance "$?s$", representing the *"SideWalk"*. Additionally, the statement "$swrlb : add(?x, ?a, ?b)$" generates a variable "$?x$" that holds the summation of the values stored in "$?a$" and "$?b$", essentially calculating the sum of the *"SideWalk"* and *"Road"* widths. Similarly, "$swrlb : divide(?d, ?x, 2)$" performs the division operation and saves the result in the variable "$?d$". Consequently, within the rule body, these conditions enable us to retrieve the widths of the *"Road"* and *"SideWalk"* concepts from the ontology and execute the necessary operations to define the distance between the sidewalk and the road center. Subsequently, the extracted distance information is attributed to the *"hasCenterRoadDistance"* data property of the

"SideWalk" concept through the rule head.

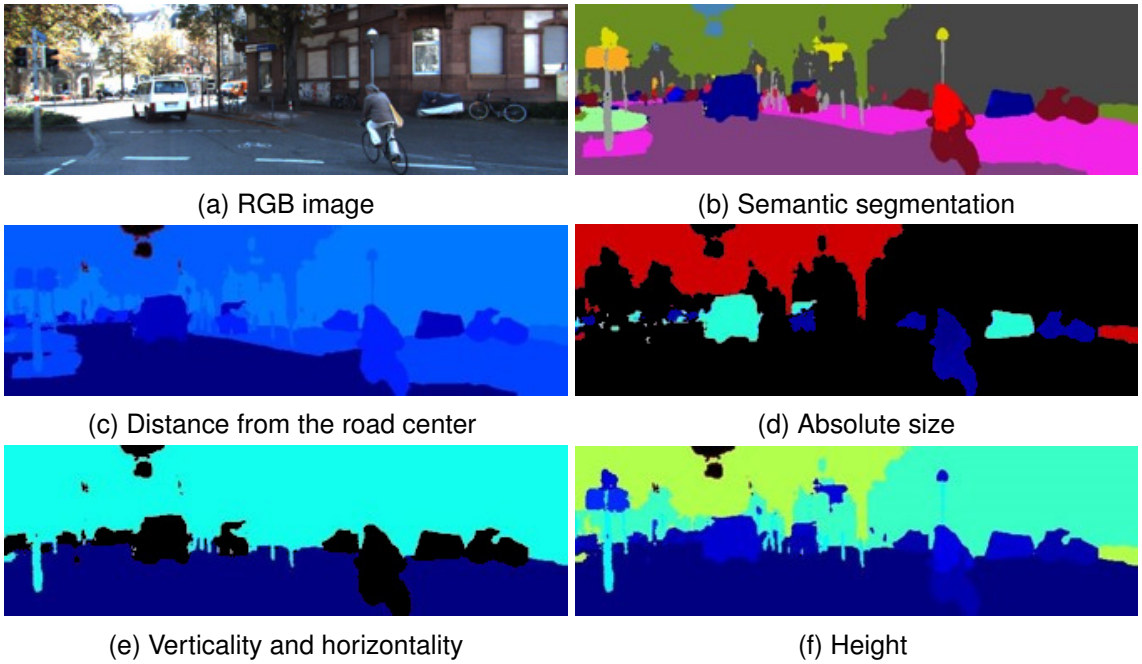
While developing the "distance from the road center" map, we also performed alternative forms of reasoning, primarily driven by the objects properties of the considered concepts. Consider again the example of the *"isOn"* relationship that establishes a connection between the *"Pedestrian"* concept and the *"SideWalk"* concept. In this scenario, when we want to deduce the ontology-based distance from the road center of a pedestrian, there is no need to recalculate it based on the pedestrain specific data properties. Instead, we can leverage the object property *"isOn"*, which inherently yields information regarding the equality of distances from the road center for both the *"SideWalk"* and the *"Pedestrian"*. The rule governing this reasoning can be summarized as follows.

$$\text{Human}(?h) \wedge \text{isOn}(?h,?i) \wedge \text{hasCenterRoadDistance}(?i,?d) \rightarrow \text{hasCenterRoadDistance}(?h,?d)$$

It is important to note that this rule is not limited only to the context of *"Pedestrian"* and *"SideWalk"*. It extends its functionality to all concepts that are part of the generic concept *"Human"*, which has a *"isOn"* relationship with another concept (see Figure 3.2). Considering similar ontology reasoning, we have applied the same methodology to extract relevant information for the creation of the remaining monocular cues maps. These monocular cues maps are briefly outlined below.

Absolute size The concept of an object absolute size significantly contributes to the depth perception process. In practical terms, this monocular cue enables us to measure the distance of an object based on its size relative to its surroundings. In the human context, our familiarity with an object size directly influences our depth perception. For instance, while driving, our knowledge of a typical car size helps in defining the positions of other vehicles on the road. Figure 3.4d serves as an example of the absolute size maps, with information in each pixel extracted from ontology reasoning. This information represents the absolute size of objects within the urban driving environment.

Verticality and horizontality The *"Linear perspective"* is a monocular cue in which parallel lines appear to converge at a distant point known as the "vanishing point". For example, when driving, the edges of a road, which are parallel, seem to approach each other until they appear to meet, even though parallel lines do not interest by definition. To help the DNN in learning the linear perspectives of urban scene components, we have represented the vertical and horizontal orientation of urban environment objects in relation to the road. Figure 3.4e represents an example of this map.



(a) RGB image (b) Semantic segmentation

(c) Distance from the road center (d) Absolute size

(e) Verticality and horizontality (f) Height

Figure 3.4: RGB image, semantic segmentation, and monocular cues maps built from ontology reasoning. A color map is used for visualization purposes, where warmer colors indicate higher information values within each pixel. In Figure 3.4d, the red color, denoting the warmest tone on the map, is allocated to pixels belonging to the vegetation class since it has the largest absolute size compared to the other scene objects. Conversely, black pixels denote classes that do not have associated monocular cues.

Height With the road as a reference point, the “height" map represents the height of objects in the urban scene with respect to the road. Height information about objects also plays a role in influencing depth perception. Consequently, if we consider two objects of the same type located at different positions, the farthest object from a reference point will appear shorter than the closest object when viewed from the same standpoint. Even in the field of art, artists simulate this effect, artists simulate this effect by representing distant objects as both smaller and shorter within the scene perspective. Figure 3.4f represents an example of the height monocular map.

3.5/ Deep neural network for monocular depth estimation

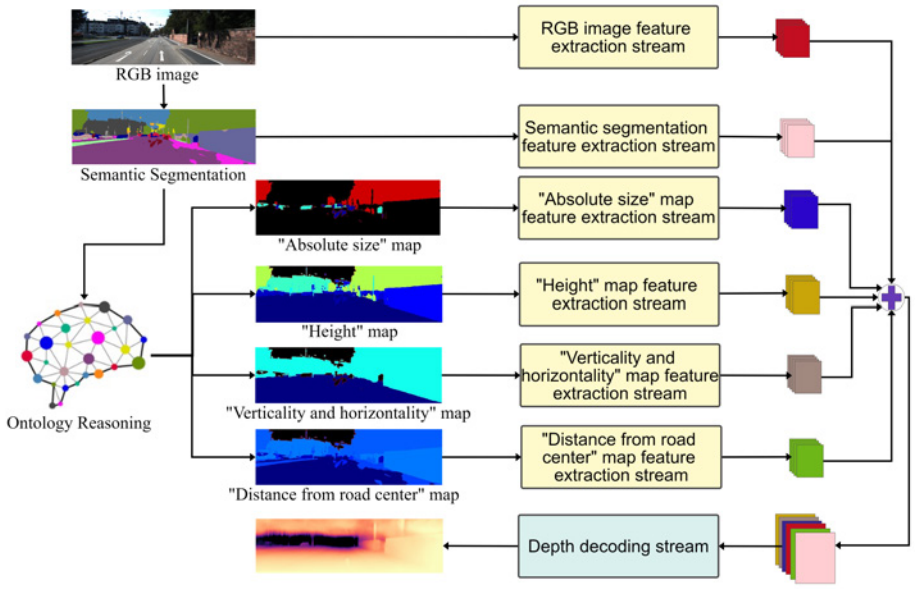
The monocular cues maps, generated through ontology reasoning, are integrated into a DNN for depth estimation, alongside the RGB image. In this context, we explore the validation of the proposed approach using two distinct models. The first model is a basic DNN, constructed based on a ResNet autoencoder architecture [31]. This choice aims to

examine the influence of monocular cues maps on the performance of a typical state-of-the-art network. The second considered DNN is the AdaBins model [136], recognized as the state of the art in supervised MDE. The aim of considering this model is to leverage a high-performing state-of-the-art model that exclusively relies on the monocular image for depth estimation and to feed it with the proposed monocular cues maps to further validate the idea of our contribution. It is worth noting that we conducted a complete and integral analysis of the ResNet-based approach. This analysis includes training and evaluation on two benchmark datasets, evaluation on the unseen dataset, and evaluation of the proposed monocular cues maps impact. This extensive evaluation was possible due to the light nature of resNet architecture and its minimal computational resource requirements. In contrast, the experiments on AdaBins, a significantly heavier and computationally intensive model, remain basic including training and evaluation specifically on the KITTI Eigen split dataset.

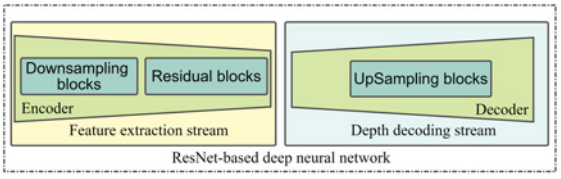
3.5.1/ MULTISTREAM PIPELINE

Given the variety of inputs, including the RGB image, semantic segmentation map, and monocular cues maps, intended for integration into the DNNs, our architecture is thoughtfully designed as a multistream pipeline. This design allows each input to be independently injected into the network, as illustrated in Figure 3.5a. The main goal behind adopting this approach is to preserve the quality of the potentially valuable information encapsulated within each input, thereby enhancing MDE results. In practice, each input is injected into the DNN in parallel with the others, initiating the feature extraction process. Once these features are extracted, they are aggregated through an element-wise sum operation. The combined filters are then fed into the final depth decoding stream, which generates the depth map. For a more in-depth exploration of the distinct blocks constituting the ResNet and AdaBins models, refer to the next paragraphs.

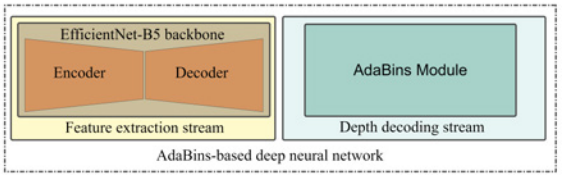
ResNet-based deep neural network The first considered DL model has a ResNet-based autoencoder architecture. This choice is based on its simplicity and widespread use in the research community [145] [146]. Among the various ResNet configurations, studies such as [147] have demonstrated that ResNet versions with 6 and 9 blocks consistently provide good performances. Additionally, the performed experiments in [147] work have shown that these variants are the most optimal. It is worth noting that using too many blocks can introduce training challenges, such as the vanishing gradient problem. As illustrated in Figure 3.5b, each input feature extraction stream includes an encoder block, which is composed of a series of downsampling and residual blocks. Subsequently, the output filters from all encoders are aggregated to feed the depth decoding stream. This component acts as the decoder and is based on upsampling blocks.



(a) General multistream pipeline architecture



(b) ResNet-based deep neural network blocks



(c) AdaBins-based deep neural network blocks

Figure 3.5: Deep neural network architecture for monocular depth estimation: (a) an overall view of the multi-stream pipeline, (b) the feature extraction stream, and depth decoding blocks for the ResNet-based deep neural network, and (c) the feature extraction stream and depth decoding blocks for the AdaBins-based deep neural network.

AdaBins-based deep neural network The second considered DL model to validate our approach is the state-of-the-art model of supervised MDE: AdaBins [136]. We have kept the core architecture of the AdaBins model unchanged, including its blocks and components. However, to accommodate the multiple inputs we intend to introduce into the model, we have organized the architecture into a multistream pipeline. As represented in Figure 3.5c, each input feature extraction stream uses the EfficientNet-B5 backbone [148], which follows an encoder-decoder architecture and is trained on the ImageNet dataset [20]. Following the same approach, the output filters are aggregated to feed the depth decoding stream, i.e., the AdaBins module.

3.6/ EXPERIMENTS AND RESULTS

To validate and demonstrate the effectiveness of the proposed methodology, we present a series of experiments in this section. These experiments aim to illustrate the advantages

and value of incorporating ontology knowledge into both a traditional DNN (ResNet) and a state-of-the-art model (AdaBins) for depth estimation.

3.6.1/ Implementation details

In the context of knowledge modeling, as discussed in Section 3.4.1, our choice for conceptualizing the ontology led us to the use of Protégé software [149]. This software offers a robust platform for modeling ontology concepts, defining their properties, and establishing relationships among them. It further provides a wide range of options for extracting the ontology in various languages. In our case, we chose to extract the ontology in OWL (Web Ontology Language) language [43] to represent the knowledge in a structured format. This language allows us to seamlessly load the knowledge and ontology rules into the Python environment using the OwlReady2 python library [150]. The adoption of OWL serves not only to structure the information but also to enable the practical integration of the proposed ontology within Python. When it comes to the ontology rules, we used the Semantic Web Rule Language (SWRL)[144].

The implementation of the ResNet model was performed using Pytorch 1.7 Neural Network Libraries, with the support of CUDA GPU Toolkit 11 for enhanced computational efficiency. The training was conducted with the Adam optimizer, using a learning rate of 0.0001. The training process was executed on a high-performance system, the Alienware Aurora R11 i9-10900KF equipped with Dual RTX2080Ti, each boasting 22GB of 22GB VRAM. To ensure a comprehensive training process, the model was trained for 20 epochs, and a batch size of 8 was considered. As for the inference computational time, the processing of a single image is performed in about 0.059s, equivalent to over 16 Frames Per Second (FPS).

The implementation of AdaBins was carried out using Pytorch 1.7 Neural Network Libraries with the support of CUDA GPU Toolkit 11 for optimal GPU acceleration. The official codebase for AdaBins is publicly accessible [151]. In our implementation, we preserved the hyperparameters as specified by the original authors. However, it is worth noting that the AdaBins model, as described in the original paper, was trained on a machine with four NVIDIA V100 GPUs, each equipped with 32GB of VRAM. The training setup featured a batch size of 16 and was executed for 25 epochs. In terms of inference, the AdaBins model processes a single image in approximately 0.448 seconds, which is equivalent to an effective frame rate of 2 FPS. Due to the difference in our hardware setup, with a single NVIDIA GPU boasting 24 GB of VRAM, we encountered constraints in using the same batch size. To address this, we adopted the gradient accumulation technique [152] in which the final step of the training process was adjusted. Instead of updating the network weights after each batch, we saved the gradient values and aggre-

gated them over a set number of batches before performing weight updates. In our case, we processed 16 batches before updating the model weights. This approach allowed us to simulate a batch size of 16 while accommodating the computational performances of our machine.

Finally, to extract the semantic segmentation necessary for generating the proposed monocular cues maps from the acquired RGB images, we considered the semantic segmentation model outlined in [153]. The model, a Feature Pyramid Network (FPN), is based on a ResNet backbone that had been pretrained on the CityScapes dataset [154]. This pretrained network achieved a mean Intersection over Union (mIoU) score of 75% on the validation set, as reported in [155]. The model enables segmenting 19 primary classes within the urban environment. It is important to highlight that the weights of this semantic segmentation network remained fixed and were not subject to updates during the training of the DNNs for MDE.

3.6.2/ Datasets and evaluation metrics

3.6.2.1/ KITTI dataset

We used the KITTI Vision benchmark suite [156] as the primary dataset for both training and evaluation. The choice of this dataset was based on its widespread use for depth estimation tasks, allowing for fair and meaningful comparisons with state-of-the-art approaches. More precisely, we demonstrated the effectiveness of our approach on the KITTI Eigen split [157]. We chose this split as it represents the most commonly used benchmark for the evaluation of MDE models [158; 136]. Consequently, we could directly evaluate the impact of integrating monocular cues maps in the proposed and considered DNNs. The training subset of this dataset includes approximately 23K RGB images, each paired with its corresponding ground truth depth map. Additionally, 697 samples are devoted to the test.

3.6.2.2/ CityScapes dataset

For a comprehensive evaluation, we extended our experimentation to include the CityScapes dataset [154]. This dataset includes a training set with 2975 images with their respective disparity maps ground truth. Afterward, 1525 images are dedicated for testing. We chose this dataset because it offers a diverse range of urban environments in different conditions compared to the KITTI one, which allows us to generalize the effectiveness of our approach. Furthermore, some state-of-the-art models [159; 160] also referenced the CityScapes dataset in their evaluations, which enables us to compare our approach on two benchmarks for outdoor scenarios.

3.6.2.3/ APPOLLOSCAPE DATASET

Lastly, we considered the AppIloScape dataset for autonomous driving [161] to evaluate the performance of our models on unseen driving data. This dataset includes 1300 images and their corresponding ground truth depth maps. The AppolloScape dataset captures data from four distinct regions in China with a range of different times of day and weather conditions, enhancing the dataset diversity which is suitable for evaluating our models performance.

3.6.2.4/ EVALUATION METRICS

For the evaluation of our results, we used common metrics for evaluating the quality of depth estimation [136; 162]. To quantitatively measure the performance of our models, we calculated four key error metrics that compare the predicted depth map to the ground truth one, in line with the definitions in [157]. These metrics include the Absolute Relative error (AbsRel), Squared Relative error (SqRel), Root Mean Square Error (RMSE), and log mean square error (logRMSE). Their specific equations are defined as follows.

$$AbsRel = \frac{1}{N}\sum_{i=0}^{N-1}\frac{|d_i - \overline{d_i}|}{\overline{d_i}} \tag{3.1}$$

$$SqRel = \frac{1}{N}\sum_{i=0}^{N-1}\frac{\left(d_i - \overline{d_i}\right)^2}{\overline{d_i}} \tag{3.2}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(d_i - \overline{d_i}\right)^2} \tag{3.3}$$

$$logRMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(\log d_i - \log \overline{d_i}\right)^2} \tag{3.4}$$

With N being the number of test images, d_i the predicted depth, and $\overline{d_i}$ the depth of the ground truth. Additionally, we computed three accuracy metrics, which give the fraction τ of predicted depth values inside an image whose ratio and inverse ratio with the ground truth are below the thresholds 1.25, 1.25^2 and 1.25^3. They are, respectively, defined as follows:

$$\tau < 1.25 : \frac{1}{N}\sum_{i=0}^{N-1}\left[max\left(\frac{d_i}{\overline{d_i}}, \frac{\overline{d_i}}{d_i}\right) < 1.25\right] \tag{3.5}$$

$$\tau < 1.25^2 : \frac{1}{N}\sum_{i=0}^{N-1}\left[max\left(\frac{d_i}{\overline{d_i}}, \frac{\overline{d_i}}{d_i}\right) < 1.25^2\right] \tag{3.6}$$

$$\tau < 1.25^3 : \frac{1}{N}\sum_{i=0}^{N-1}\left[max\left(\frac{d_i}{\overline{d_i}}, \frac{\overline{d_i}}{d_i}\right) < 1.25^3\right]. \tag{3.7}$$

3.6.3/ Training and testing process

Throughout the training phase, we used the traditional Least Absolute Deviation Function, denoted as $L1$, to compute the error between the ground truth depth map and the predicted one at each iteration. The $L1$ equation is defined as follows.

$$L1 = |d_i - \overline{d_i}| \tag{3.8}$$

During the testing phase, the trained model was supplied with both the RGB images and their corresponding monocular cues maps. Subsequently, we obtained the predicted depth map, which was compared directly with the ground truth to evaluate the model performance. We decided to use the $L1$ loss because the main goal of the proposed approach is to demonstrate the added value of monocular cues extracted from ontologies on a simple and basic model in terms of architecture and learning process. Therefore, we fixed all the elements other than the injected monocular cues that can enhance the model accuracy such as the loss function, the model layers, etc. Regarding the AdaBins model, the same training and testing protocol described in [136] was followed.

3.6.4/ Experiments and evaluation of ResNet-based deep neural network

This section includes two main parts. First, we present the results of our experiments with the ResNet model on two different datasets (Section 3.6.4.1 and Section 3.6.4.2). In the first part, we focus on the analysis and discussion of results obtained through the proposed approach. Subsequently, we provide a comparative analysis with state-of-the-art models in Section 3.6.4.4.

3.6.4.1/ Evaluation on KITTI Eigen split

To validate our approach, we report in Table 3.1 “Proposed approach (supervised)" the results of four experiments that consisted of training and evaluating the ResNet model according to different subsets of inputs, mainly:

- (M): single use of the monocular image,
- (M+Sem): monocular image + semantic segmentation,

- (M+Sem+Onto): monocular image + semantic segmentation + the 4 proposed monocular cues maps extracted from ontology reasoning,
- (M+Onto): monocular image + the 4 proposed monocular cues maps extracted from ontology reasoning.

We would like to remind you that the primary aim behind considering the ResNet architecture is to evaluate the influence and impact of monocular cues maps on a typical and basic state-of-the-art network. The central focus of this section is to study the impact of the monocular cues maps independently, without drawing comparisons with state-of-the-art models. However, the discussion and the comparison with the state of the art are presented in Section 3.6.4.4.

Based on the results of the initial experiment (M), it is evident that the exclusive use of the monocular image yields promising results in terms of accuracy (95% for the third threshold) and also shows minimal errors (with Abs Rel at 0.184). This validates the assumption that DL models implicitly leverage additional information from monocular images during their learning process for depth estimation, as previously suggested in [133]. In the second experiment, involving the inclusion of semantic segmentation during the ResNet training (M+Sem), considerably better results are achieved compared to the first experiment (M), especially in terms of accuracy. In this context, we observe an improvement of 5%, raising the accuracy from 95.0% to 97.1%. This outcome validates the idea that providing the model with additional information about the semantic attributes of objects significantly helps DNNs in MDE. It is essential to understand that the goal of this experiment (M+Sem) is to define a base for validating and evaluating the impact of incorporating monocular cues maps explored in the next experiment (M+Sem+Onto). Therefore, the improved results observed when comparing the experiments (M+Sem) and (M+Sem+Onto) can be attributed to the additional ontology knowledge injected into the model, rather than the semantic information.

The experiment (M+Sem+Onto), which involves the simultaneous injection of the RGB image, the semantic segmentation, and the monocular cues maps, stands as the core of the approach presented in this chapter. As evident in Table 3.1, the results obtained from this experiment (M+Sem+Onto) outperform the results from the other experiments (M) and (M+Sem), in terms of accuracy across all thresholds. In this regard, the accuracy for the first threshold, considered the most strict, reaches 89.1% (M+Sem+Onto) compared to 86.2% for the experiment (M+Sem). Furthermore, a reduction in all error metrics is observed, indicating that the model fits the dataset more effectively. This performance was expected because the knowledge transfer through the monocular cues maps provided the depth estimation model with additional insights about the urban environment and its components. Consequently, the network could explicitly and directly acquire the required cues for depth estimation. Without adding the monocular cues maps, the results

Method	Depth Estimation Evaluation						
Model	*Errors*				*Accuracy*		
	AbsRel	***SqRel***	***RMSE***	***logRMSE***	$\tau < 1.25^1$	$\tau < 1.25^2$	$\tau < 1.25^3$
State-of-the-art supervised methods							
Saxena et al. [158]	0.280	3.012	8.734	0.361	0.601	0.820	0.926
Liu et al. [163]	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Eigen et al. [157]	0.203	1.548	6.307	0.282	0.702	0.898	0.967
Kuzniestov et al. [164]	0.122	0.763	4.815	0.194	0.845	0.957	0.987
Gurram et al. [165]	0.100	0.601	4.298	0.174	0.874	0.966	0.989
Gan et al. [166]	0.098	0.666	3.933	0.173	0.890	0.964	0.985
Fu et al. [146]	0.072	0.307	2.727	0.120	0.932	0.984	0.994
Yin et al. [167]	0.072	–	3.258	0.117	0.938	0.990	0.998
Song et al. [162]	0.059	–	2.446	0.091	0.962	0.994	0.999
Bhat et al. [136]	0.058	0.190	2.360	0.088	0.964	0.995	0.999
Proposed approach (supervised)							
ResNet (M)	0.184	0.903	3.602	0.153	0.843	0.928	0.950
ResNet (M+Sem)	0.165	0.701	3.104	0.131	0.862	0.953	0.971
ResNet (M+Sem+Onto)	0.098	0.572	2.791	0.118	0.891	0.964	0.986
ResNet (M+Onto)	0.094	0.551	2.775	0.117	0.891	0.964	0.986
State-of-the-art self-supervised and unsupervised methods							
Godard et al. [134]	0.132	1.044	5.142	0.210	0.845	0.948	0.977
Gur et al. [168]	0.110	0.666	4.186	0.168	0.880	0.966	0.988
Bian et al. [169]	0.137	1.089	5.439	0.217	0.830	0.942	0.975

Table 3.1: Comparison with state-of-the-art models on KITTI Eigen Split. “M”, “Sem” and “Onto” respectively refer to the ResNet model trained using monocular images, semantic segmentation, and the four proposed monocular cues maps extracted from ontology reasoning.

(M+Sem) show that trying to explore monocular cues only from the RGB image and its semantic map is less accurate and leads to higher errors.

The validation of our approach through the final experiment (M+Sem+Onto) inspired us to conduct another experiment that consists of excluding the semantic segmentation map. This time, the model was trained only using the RGB image and the monocular cues maps. The results (M+Onto) from this experiment are approximately equivalent to those of the prior experiment (M+Sem+Onto) in terms of accuracy. However, considering the error metrics, the results improved slightly. Consequently, training the model using only the RGB image and the monocular cues maps, without the inclusion of the semantic segmentation map, appears to be a more efficient approach. This confirms our initial assumption, or at least one can say that the semantic map is not needed when the monocular cues are included, which guides the model to a shorter training path, resulting in a lower error rate.

3.6.4.2/ EVALUATION ON CITYSCAPES DATASET

Our choice to conduct experiments on the CityScapes dataset involved using three distinct semantic segmentation models, each with varying performance levels: DeepLabV3 [170], ICNet [171] and Fast-SCNN [170]. This selection was made to later evaluate the influence of the semantic segmentation accuracy on the proposed approach. These models were trained on the CityScapes dataset, resulting in respective mIoU scores of $79,4\%$, 74.5%, and $72,3\%$.

The results obtained, as presented in Table 3.2, reaffirm the significant contribution of the proposed monocular cues maps towards enhancing depth estimation accuracy. In this context, the performance levels improved from 85.6% (M+Sem) to 87.7% (M+Sem+Onto), particularly when employing the best-performing semantic segmentation model, DeepLabV3. Furthermore, our ability to reduce the model error (Abs Rel) from 0.119 (M+Sem+Onto) to 0.111 (M+Onto) by excluding the semantic segmentation is evident.

Semantic segmentation		**Inputs**	**Depth Estimation Evaluation**			
Proposed approach						
Model	***mIoU***	***M/Sem/Onto***	***Error***	***Accuracy***		
			Abs Rel	$\tau < 1.25^1$	$\tau < 1.25^2$	$\tau < 1.25^3$
DeeplabV3 [170]	79.40%	M+Sem	0.134	0.856	0.951	0.976
		M+Sem+Onto	0.119	0.876	0.957	0.980
		M+Onto	0.111	0.877	0.957	0.980
ICNet [171]	74.50%	M+Sem	0.130	0.857	0.952	0.977
		M+Sem+Onto	0.119	0.874	0.956	0.979
		M+Onto	0.112	0.874	0.956	0.979
FastSCNN [170]	72,30%	M+Sem	0.140	0.834	0.944	0.973
		M+Sem+Onto	0.125	0.874	0.954	0.978
		M+Onto	0.119	0.873	0.954	0.978
State-of-the-art methods						
Wang, Lijun, et al. [159] (S)			0.227	0.801	0.913	0.950
Laina, Iro, et al. [145] (S)			0.257	0.765	0.893	0.940
Xu, Dan, et al. [172] (S)			0.246	0.786	0.905	0.945
Zhang, Zhenyu, et al. [173] (S)			0.234	0.776	0.903	0.949
Saeedan, Faraz, et al. [160] (S-S)			0.178	0.771	0.922	0.971

Table 3.2: Comparison with state-of-the-art methods on CityScapes. Results of [145; 172; 173] were implemented and evaluated by [159] and [160].

On the other hand, we find that the more performant the semantic segmentation model (higher mIoU), the better the depth estimation results. This mainly comes down to the fact that semantic segmentation is used to carry out the mapping between the classes assigned to the objects of the urban environment and the concepts of ontology. Consequently, the quality of the monocular cues maps extracted from ontology benefits from the improved semantic segmentation model.

3.6.4.3/ EVALUATION ON UNSEEN DATASET

To perform a comprehensive evaluation of our model performance, we went one step further and analyzed how well our ResNet-based approach generalizes to an unseen dataset. To this end, we evaluated on the AppolloScape dataset for autonomous driving [161]. More specifically, we inferred our model on 1300 images from the considered dataset and proceeded to compare the results with their respective ground truth depth maps. We performed this experiment with the following models: ResNet model pretrained on KITTI and ResNet model pretrained on CityScapes. Furthermore, we used the best semantic segmentation model (DeepLabV3) for the experiment (M+Sem+Onto) considering the ResNet model pretrained on CityScapes.

The results shown in Table 3.3 provide more evidence that the monocular cues maps extracted from the ontology reasoning contribute to improving the overall model performance. This can be seen from the enhanced accuracy results with the experiment (M+Onto) that reached 46.2% for the first threshold, compared to the 44.9% achieved by the experiment (M+Sem) for the model trained on KITTI. The same effect is observed when applying the ResNet-based model pretrained on CityScapes and evaluating it on the AppolloScape dataset.

Method	***Accuracy***		
	$\tau < 1.25^1$	$\tau < 1.25^2$	$\tau < 1.25^3$
Model trained on KITTI			
ResNet (M)	0.441	0.836	0.904
ResNet (M+Sem)	0.449	0.845	0.913
ResNet (M+Sem+Onto)	0.460	0.847	0.918
ResNet (M+Onto)	0.462	0.847	0.919
Model trained on CityScapes			
ResNet (M)	0.450	0.849	0.921
ResNet (M+Sem)	0.461	0.853	0.940
ResNet (M+Sem+Onto)	0.472	0.861	0.948
ResNet (M+Onto)	0.472	0.862	0.948

Table 3.3: Evaluation of the ResNet-based approach on unseen scenarios from the AppolloScape dataset.

3.6.4.4/ COMPARISON WITH THE STATE OF THE ART

To evaluate the effectiveness of our ResNet-based model, we conducted a comparative analysis against state-of-the-art models for MDE, considering two key benchmarks: KITTI Eigen split and CityScapes. It is important to note that we did the comparison with approaches that have proposed models trained with a supervised pattern. However, for a more comprehensive view and a larger perspective, we have also reported the best models for unsupervised and self-supervised approaches.

Comparison with the state-of-the-art models trained and evaluated on KITTI Eigen split We can notice from the results of the Table 3.1, that our ResNet-based approach outperformed some of the leading supervised models in depth estimation, such as [166] and all the earlier models. However, more recent models, such as [174] or [136], which are currently considered the state of the art in the target task, obtained more performant results, especially for the first threshold. We can say that this result is logical and expected since our model is based on an original ResNet autoencoder without any modification of its architecture. Furthermore, the state-of-the-art works propose approaches that aim to improve the performance of the neural networks with several techniques such as the modification of layers, the proposition of new loss functions, etc. For this reason, experiments on a more efficient model have also been performed and will be discussed in Section 3.6.5.

Comparison with the state-of-the-art models trained and evaluated on CityScapes The top results of the state of the art in MDE as well as those provided by our approach, all evaluated on CityScapes, are reported in Table 3.2. The proposed approach outperformed all the state-of-the-art models, regardless of the semantic segmentation model used. One can see that our method reached an accuracy of 98% for the third threshold using DeepLabV3 as a semantic segmentation model. Considering the state of the art, the best supervised model [159] reached 95% accuracy, and 97.1% was obtained regarding the best model based on self-supervised training pattern and using semantic and instance segmentation [160].

3.6.4.5/ Ablation Study

Additional experimentations were performed to evaluate the influence of individual monocular cues maps on the depth estimation outcomes. This evaluation includes all the possible combinations as shown in Table 3.4. It is important to note that all the experiments mentioned in Table 3.4 exclude the semantic segmentation map and include the monocular image (+M) as input to the model.

Our ResNet-based model was first trained based on the KITTI dataset, with dedicated experiments considering each of the proposed monocular cues maps separately. The findings, as presented in the four first rows of Table 3.4 (designated as MC#1) represent experiments that consider only one monocular cue. The results highlight the influence of objects absolute size knowledge on enhancing depth estimation outcomes.

Experiment #1 which exclusively uses the absolute size map stands out with the highest accuracy among these individual experiments when compared to the remaining monocular cues maps tested independently in experiments #2, #3, and #4. The second-best re-

MC#	Exp#	Model inputs (+M)				Accuracy results		
		Abs. size	Height	Dist. from RC	Vert. and horiz.	$\tau < 1.25^1$	$\tau < 1.25^2$	$\tau < 1.25^3$
1	1	X				0.855	0.941	0.983
	2		X			0.841	0.923	0.974
	3			X		0.736	0.897	0.971
	4				X	0.704	0.877	0.954
2	5	X	X			0.851	0.930	0.978
	6	X		X		0.858	0.949	0.983
	7	X			X	0.852	0.940	0.983
	8		X	X		0.849	0.929	0.978
	9		X		X	0.847	0.926	0.977
	10			X	X	0.739	0.899	0.957
3	11	X	X	X		0.858	0.950	0.984
	12	X	X		X	0.857	0.946	0.983
	13	X		X	X	0.891	0.963	0.986
	14		X	X	X	0.888	0.960	0.985
4	15	X	X	X	X	**0.891**	**0.964**	**0.986**

Table 3.4: Evaluation of the monocular cues maps impact on our ResNet-based model using KITTI dataset. “Abs. size", “Height", “Dist. from RC" and “Vert. and horiz." refer to the four proposed monocular cues maps. “MC" refers to the number of monocular cues maps included in the combinations of each experiment block.

sults among these four first experiments were obtained by training the model only with the height map. These results confirm the hypothesis that DL models implicitly use height information to estimate depth [132]. We also noticed that the absolute size map performed better results than the height map because the object absolute size information includes its height, so obviously the first map contains the knowledge of the second one with additional geometric information. The results provided by the distance from the road center map reported in experiment #3 or the verticality and horizontality map in experiment #4 are promising but less interesting than the two remaining maps, since the accuracy with the first threshold is approximately 10% lower compared to the other maps regarding experiments #1 and #2. Indeed, these two maps have been inspired by the concepts of monocular cues and do not represent them directly.

The experiment results reported in rows 5 to 10 (designated as MC#2) represent all the possible combinations of two monocular cues maps. The majority of these combinations: experiments #6, #8, #9, and #10, show that the addition of a second monocular cue map improves the results when compared to the integration of a single map. However, the result reported in experiment #6 indicates that the addition of the verticality and horizontality map to the absolute size map has almost no effect on the result compared to the accuracy obtained through experiment #1, which relies on the single integration of the absolute size map. Furthermore, experiment #5 shows that the combination of the absolute size map and height map is also not performing well, which is probably because the information included in the height map is redundant.

Finally, experiment results with a combination of 3 monocular cues maps are reported in rows 11 to 14 (designated as MC#3). Mainly, we can say that the combinations adopted in the experiments #11, #12, and #14 have a significant impact on the results. On the other hand, if we compare the results of experiment #13 with #15, which represents the combination of the four proposed maps, we can say that the height map brings only a minor improvement in the performance.

According to the above results, one can say that the association of the four proposed monocular cues maps leads to obtaining the best results. This association makes them cooperate to obtain a strong model for depth estimation based on different knowledge complementing each other. We also performed the same experiment considering the CityScapes dataset and obtained results leading to the same conclusion.

3.6.5/ Experiments and evaluation of AdaBins-based deep neural network

To validate and confirm the effectiveness of our proposed approach, we conducted an evaluation using the AdaBins model, which stands as the leading state-of-the-art model in MDE. This evaluation involved training the AdaBins model according to our approach, wherein we incorporated the ontology-based monocular cues maps with the RGB image. This integration was achieved using a multistream pipeline, aligning with the approach we adopted for the ResNet-based model.

3.6.5.1/ Evaluation on KITTI Eigen split

The results presented in Table 3.5 report the different experiments performed on the AdaBins model considering the KITTI Eigen split. For reference only, we have reported in the first row the results published in [136] paper. The result reported in the second row of Table 3.5, shows a gap in terms of performance compared to the paper results [136]. This is due to the difference between the specifications of our machine and the one used in [136] for the training; more details have been mentioned in the implementation details (Section 3.6.1). Therefore, the other performed experiments including the monocular cues maps (Onto) and semantic segmentation (Sem) are compared with the AdaBins results obtained with our local implementation (Table 3.5, row 2).

According to the results obtained and reported in Table 3.5, we can notice an improvement in terms of accuracy with the addition of extra knowledge and information. We first experimented with adding semantic segmentation to the RGB image (M+Sem). As expected, the results were improved from 88.3% to 90.1% considering the first threshold. Concerning the experiment (M+Sem+Onto), we identify once again a significant improve-

Model	Errors				Accuracy		
	Abs Rel	***Sq Rel***	***RMSE***	***RMSE log***	$\tau < 1.25^1$	$\tau < 1.25^2$	$\tau < 1.25^3$
AdaBins (Paper) “M" [136]	0.058	0.190	2.360	0.088	0.964	0.995	0.999
AdaBins (Our Impl.) “M"	0.104	0.470	3.457	0.128	0.883	0.973	0.992
AdaBins “M+Sem"	0.101	0.352	3.241	0.120	0.901	0.982	0.993
AdaBins “M+Sem+Onto"	0.095	0.328	3.109	0.115	**0.923**	**0.983**	**0.993**
AdaBins “M+Onto"	**0.089**	**0.310**	**3.008**	**0.112**	0.922	**0.983**	**0.993**

Table 3.5: Performances on AdaBins-based deep neural network in the KITTI Eigen Split against the baseline. The first row represents the AdaBins results reported in the official paper [136] for context.

ment in the model performances. The accuracy results have increased according to the three thresholds. For example, an enhancement of 2.2% has been observed about the first threshold. However, there is only a slight improvement considering the second and third thresholds since the results obtained only with the RGB image concerning the same thresholds are almost saturated, reaching more than 98%.

Concerning the error metrics which also show the performance of the model and its learning abilities, we can say that the monocular cues maps contribute to lowering the error rates. For example, the Abs Rel error reached a value of 0.095 following the experiment (M+Sem+Onto) instead of 0.101 considering the experiment (M+Sem). Finally, the last experiment excluding the semantic segmentation (M+Onto), further confirms that the use of the proposed monocular cues maps with the RGB image provides very good results. There is mainly no difference considering the accuracy results. However, a slight improvement can be noticed in error evaluation metrics. Consequently, this configuration is once again the most advantageous, as we consider one less input (Sem) in the training of our model while keeping the best performances.

3.6.5.2/ Evaluation on unseen dataset

To evaluate the performance of our AdaBins-based approach on an unseen dataset, we evaluated our pretrained model on KITTI Eigen Split according to the different proposed configurations: (M), (M+Sem), (M+Sem+Onto) and (M+Onto) on the AppoloScape dataset. The results, as presented in Table 3.6, reaffirm our initial assumption, claiming that the monocular cues maps, extracted from the ontology reasoning, indeed enhance the performance of the DNN, even in the unseen scenario case. The improvement in the accuracy of the model can be identified, especially for the strictest threshold. The addition of ontology-based knowledge allowed us to achieve an improvement from 56% (M+Sem) to 57% (M+Sem+Onto) in an unseen environment, which is encouraging and promising regarding the proposed approach. Concerning the last experiment (M+Onto), we were also able to keep the same accuracy while excluding the semantic segmentation,

validating again the independent contribution of the ontology knowledge injected into the model. Finally, we would like to mention that the basic experiments realized on AdaBins allowed us to validate our approach and to have a complete view of the impact of the proposed monocular cues maps on a basic model (ResNet) and a heavy and powerful one (AdaBins).

Method	***Accuracy***		
	$\tau < 1.25^1$	$\tau < 1.25^2$	$\tau < 1.25^3$
AdaBins (M)	0.552	0.900	0.923
AdaBins (M+Sem)	0.561	0.909	0.931
AdaBins (M+Sem+Onto)	0.570	**0.913**	**0.932**
AdaBins (M+Onto)	**0.571**	**0.913**	**0.932**

Table 3.6: Evaluation of our AdaBins-based model trained on KITTI Eigen split on unseen scenarios of AppolloScape dataset.

3.7/ CONCLUSION AND FUTURE WORK

This chapter introduces a novel approach for MDE that leverages knowledge of the urban environment through ontology reasoning. The proposed system [175; 176] extracts monocular cues based on reasoning performed on the proposed ontology. It contains different concepts of the urban environment as well as several geometric and spatial information representing basic human knowledge. Furthermore, all the information contained in the acquired RGB image related to the urban environment and the extracted knowledge from the ontology is fed into a DNN model as separate inputs in a multistream way. Several experiments were performed to validate and evaluate the impact of adding monocular cues maps and human-like reasoning to the depth estimation process. The proposed approach was deployed in two DL models: a basic model (ResNet) and a heavy powerful one (AdaBins). These models have been trained based on three main experimentations: taking as input the RGB image and the semantic segmentation map, conserving those two inputs and adding the monocular cues maps, and finally excluding the semantic segmentation and conserving only the RGB image and the monocular cues maps. The results have shown that a model trained based on the monocular cues maps achieves better results compared to the other performed experiments. On the other hand, experimentations performed on both seen and unseen challenging real-world datasets, show that the DNNs, trained based on the monocular cues maps extracted from ontology reasoning, consistently improve the state-of-the-art MDE results. This confirms the initial hypothesis that integrating human knowledge into a DNN has a beneficial impact on the target task. In addition, it confirms the fact that the direct integration of monocular cues as input in the DNN training process leads to faster model learning. Our future

research directions include leveraging other monocular cues, such as texture gradient, shading, and lighting, as well as performing more advanced ontology reasoning to extract other relevant information for depth estimation. Furthermore, we aim to investigate other hybrid strategies for combining knowledge with DNNs to enhance computer vision task performance. These strategies can include alternative knowledge representations and also different knowledge integration methodologies into the DNNs. This will be the focus of our next chapter, within the context of our second contribution to this thesis.

4

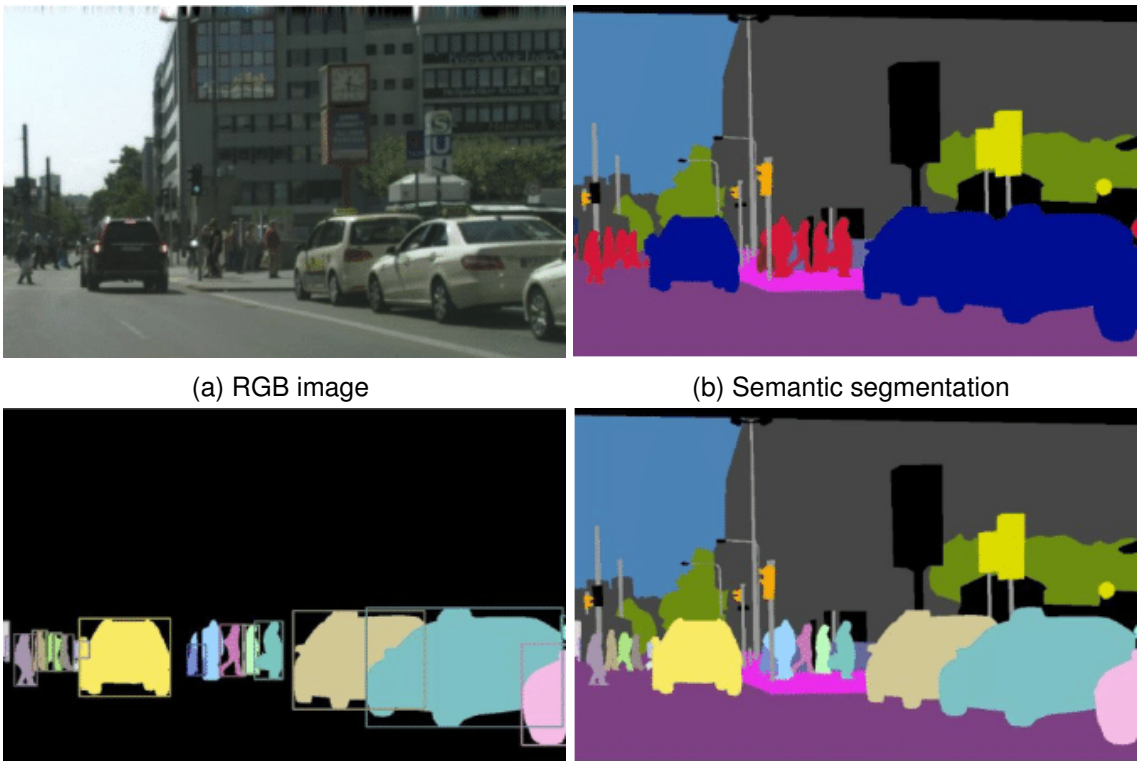
HYBRID AI FOR PANOPTIC SEGMENTATION: AN INFORMED DEEP LEARNING APPROACH WITH INTEGRATION OF PRIOR SPATIAL RELATIONSHIPS KNOWLEDGE

4.1/ INTRODUCTION & CONTEXT

In the last chapter, we successfully confirmed our hypothesis that integrating knowledge into a DNN as input to the model significantly enhances its performance. This validation was achieved specifically within the context of the MDE task. Having established the efficiency of this hybrid approach, the present chapter introduces another novel strategy. This strategy involves the integration of knowledge directly during the training loop of the model, with the primary goal of investigating how knowledge integration can further enhance the performance of a DNN. For this strategy, we have selected the panoptic segmentation task as our target application.

Panoptic segmentation, as shown in Figure 4.1d, is a computer vision task designed to recognize and categorize all elements within an image by integrating information from both semantic and instance segmentation. Semantic segmentation, as illustrated in Figure 4.1b, divides an image (Figure 4.1a) into regions associated with non-quantifiable object classes, often referred to as “Stuff", which can include elements like the sky or the road. It is also able to categorize quantifiable objects, but it does not provide individual distinction. In contrast, instance segmentation (Figure 4.1c), involves the precise identification of individual quantifiable objects in the image, referred to as “Things", such as cars or pedestrians. Panoptic segmentation ability to comprehensively describe and

analyze images offers practical solutions across a range of applications. In the domain of mobile robotics, for example, it plays a pivotal role in the detection and tracking of moving objects [177]. Furthermore, this task significantly contributes to the field of autonomous driving, empowering vehicles to gain a deep understanding of their surroundings and make precise decisions [178; 179]. Figure 4.1 illustrates a scene captured in the context of autonomous driving, along with its corresponding semantic, instance, and panoptic segmentation.



(a) RGB image (b) Semantic segmentation

(c) Instance segmentation (d) Panoptic segmentation

Figure 4.1: Panoptic segmentation of an image can be considered as a combination of semantic and instances of perceived objects.

Since 2018, there has been a growing interest within the scientific community regarding the prediction of panoptic segmentation [180]. This approach is recognized as a collaborative one that combines the strengths of both the semantic and instance segmentation methods. Panoptic segmentation techniques find common use in image data, relying on various DL-based strategies. Some of these methods involve employing distinct neural sub-networks for semantic and instance predictions [180]. However, this dual-network approach can be complex and have limitations in terms of effectiveness, often necessitating complicated post-processing to merge the associated predictions [181]. To address these limitations, a novel category of panoptic segmentation techniques has emerged, based on the use of a shared backbone [181]. These approaches enhance the training process

by facilitating the exchange of features between the semantic and instance segmentation modules through a common backbone encoder. This feature exchange significantly enhances the outcomes of panoptic segmentation. Moreover, these methods outperform alternative approaches when it comes to reducing the complexity of post-processing for panoptic prediction [181].

Previous studies have demonstrated the significant impact of contextual information and object relationships in enhancing computer vision tasks, particularly in the domain of object detection [182; 183]. These investigations have primarily used post-processing techniques to reevaluate identified objects taking into account object relationships, such as co-occurrence [184; 185; 186]. For example, certain objects, such as a sofa and a traffic sign, are not typically expected to co-exist within the same scene due to their associations with different environments, indoors and outdoors, respectively. It is worth noting that most of these studies were conducted before the widespread integration of DL techniques. Within the realm of DL research, there has been limited progress in employing object relations to enhance object detection tasks. Most current methods remain primarily focused on the recognition and identification of objects, regardless of their relationships [187]. One of the main challenges in this context lies in the complexity of modeling the spatial relations between objects, considering their potential disparities in position within an image, varying scales, and diverse shapes, to cite just a few.

On the other hand, some studies [188; 189], have demonstrated that CNNs have certain abilities to acquire contextual insights autonomously and implicitly during the training process [190]. Through the use of local receptive fields, [191], CNNs capture contextual details within small local regions connected to each neuron. As the network delves deeper, these receptive fields expand, thereby facilitating the assimilation of more extensive global contextual information. These outcomes highlight the importance of providing DNNs with explicit access to contextual information to further enhance their performance and accuracy. While DNNs can learn some level of contextual information through their architecture, the incorporation and transfer of this knowledge in a more explicit way can offer significant advantages. First, explicit integration of contextual knowledge enables the models to capture fine-grained cues that may be missed with implicit learning. Second, it helps empower models to make more accurate predictions, especially in challenging scenarios where implicit learning struggles to capture complex relationships effectively.

As deep network research continues to explore their capacity to learn contextual information, it becomes clear that further enhancing their performance and accuracy can be achieved by incorporating explicit access to contextual knowledge. This concept is aligned with the principles of hybrid intelligent systems [192; 193; 194] which aim to combine the strengths of artificial intelligence with human expertise. Within the domain of hybrid AI, an outstanding approach is informed DL [195; 196], which leverages prior

knowledge or domain expertise to improve the learning performance of DL models. This prior knowledge can take various forms, including expert rules, ontologies, and statistical information, among others as mentioned in Chapter 2. By incorporating this pre-existing knowledge, DL models can make more informed predictions and enhance the decision-making process. Informed DL represents a significant advancement in hybrid AI, as it combines the data-driven effectiveness of DL with valuable insights from human expertise and existing knowledge. This integration ultimately results in more robust and effective AI systems able to tackle complex real-world problems.

The integration of contextual information into DL models should be advantageous for computer vision tasks. Contextual information can be globally defined as the surrounding cues in the environment that provide additional insights and understanding to aid in accurate estimations and predictions. This includes considering spatial relationships between objects, which are crucial contextual information that can highly benefit object detection tasks for several reasons. First, incorporating spatial relations enables a more comprehensive understanding of the scene, as objects in the real world are not isolated, but rather interact and exist with each other. By capturing spatial relationships such as overlapping and relative positions to name just a few, DNNs can better understand the context and improve their performances. Second, object relations help to resolve ambiguities that may arise when objects share similar visual characteristics. For example, in a crowded scene where objects may occlude or partially overlap, understanding spatial relationships can help identify individual objects. By analyzing the spatial arrangement of objects, the network can differentiate between overlapping instances and assign correct labels to each object, thereby reducing confusion and improving object detection accuracy. Furthermore, in urban scenarios, objects such as traffic lights and traffic signs are often located on roads or sidewalks. By considering the spatial relationship between these objects and the road or sidewalk regions, DNNs can effectively detect and classify traffic-related objects. Moreover, in urban scenes, the sky regions are typically externally connected to vegetation or building regions since these objects are the tallest in urban environments. Learning the spatial relationship that connects the sky to vegetation and buildings can help the network understand that the sky is usually associated with tall objects. It is also important to mention that in urban scenes, some spatial relationships can serve as cues for identifying unrealistic or impossible scenarios. By analyzing the spatial connections between objects, it becomes possible to identify the presence or absence of certain configurations that are unlikely or impossible in urban environments. For example, the presence of a pedestrian region within a sky region is a configuration that would rarely, if ever, occur in reality. This spatial relationship is incoherent with the typical arrangement. By recognizing this inconsistency, DNNs can leverage spatial relationships to identify and differentiate between realistic and unrealistic configurations in urban scenes.

In this context, we have observed that panoptic prediction in urban environments is partic-

ularly challenging because of the complex relationships between regions within an image. To address this issue, the key contributions of this chapter are as follows.

- the extraction and integration of knowledge about spatial relationships into a deep neural network for panoptic segmentation,
- the modeling of the spatial relationships as a loss function to optimize the network training,
- the validation and evaluation of the proposed approach on various urban scene datasets.

To present our approach, the remainder of this chapter is organized as follows. Work related to panoptic segmentation is introduced and discussed in Section 4.2. The spatial relationships considered are described in Section 4.3. The proposed methodology that includes the modeling of spatial relationship loss functions is described in Section 4.4. Section 4.5 presents the performed experiments, results analysis, and comparison with the SOTA, ablation study, generalization capability, and quantitative analysis of the interest in integrating spatial relationships into the learning process. Finally, the last section concludes the chapter and provides directions for future work.

4.2/ SHARED BACKBONE MODELS FOR PANOPTIC SEGMENTATION

In this section, we present an overview of existing panoptic segmentation methodologies, with a specific focus on those built upon a shared backbone architecture. These techniques use a single neural network backbone for both “Stuff" and “Things" segmentation to achieve a unified panoptic segmentation of the image.

Over the years, many frameworks have been developed following different techniques for panoptic segmentation. An effective approach is to use a shared backbone to encode features [197; 198; 199; 200], as it has been shown to produce high performance on benchmark datasets [201; 154; 202]. Within this category of techniques, two primary approaches are illustrated in Figure 4.2. The first involves sharing a backbone between the two heads of semantic and instance segmentation and merges the outputs for the final panoptic generation. In addition to the shared backbone, the second category includes explicit connections between the two heads. Many methods have been proposed in the state of the art and can be classified into one of these two categories. In this section, we review some of the most important methods in each category and present their contributions to panoptic segmentation.

The approach proposed in [180] performs the instance and semantic segmentation separately and then applies the Non-Maximum Suppression (NMS) technique to obtain the

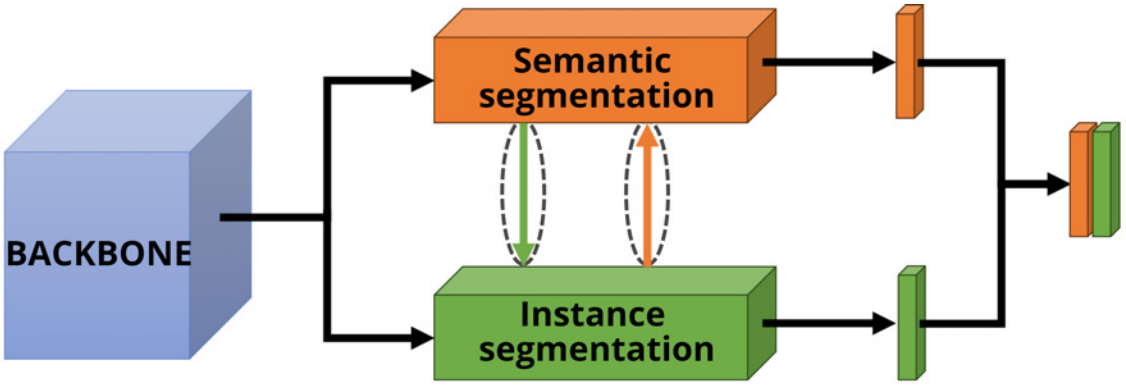


Figure 4.2: Comparison of two sharing backbone architectures for panoptic segmentation. The whole flowchart represents the architecture with a shared backbone and explicit connections. The arrows enclosed with dashed lines can be excluded to obtain the architecture without explicit connections.

Panoptic Quality (PQ) metric. The NMS procedure is used to produce non-overlapped instance regions, which are then combined with the semantic segmentation. The problem of conflicts that may arise within the instance segmentation branch due to overlapped predictions was highlighted in [203]. The proposed contribution consists of adding a branch, called the occlusion head, responsible for making decisions regarding the stacking order of instance masks to resolve occlusions during the fusion process. The Efficient Spatial Pyramid of dilated convolutions (ESPnet) was introduced in [199]. This method involves several stages, including a shared backbone that consists of a Feature Pyramid Network (FPN) [153] and a Residual Network (ResNet) [204]. To enhance the input features, the method uses a Cross-Layer Attention (CLA) fusion module, which combines multi-layer feature maps in the FPN layer. The approach proposed in [197] introduces the Efficient Panoptic Segmentation (EfficientPS) architecture for scene understanding. The general architecture of the network consists of a shared backbone that encodes and fuses semantically rich multi-scale features. It includes a new semantic head that aggregates fine and contextual features consistently. For the instance segmentation head, a new variant of Mask R-CNN [197] augmented with depth-wise separable convolutions [205] is considered. Finally, a novel panoptic fusion module is introduced to generate the final panoptic output. A new system called Panoptic-DeepLab for panoptic segmentation is presented in [198]. The approach employs a shared backbone network (Xception-71 [206]) augmented with an atrous convolution in the final block. The architecture is based on dual-Atrous Spatial Pyramid Pooling (ASPP) and dual-decoder structure specific to semantic and instance segmentation respectively. The semantic branch follows the standard design of a semantic segmentation model, while the instance branch is class-agnostic and uses a simple instance-center regression. The predicted semantic segmentation and instance segmentation are fused to generate the final panoptic segmentation result by the majority vote algorithm proposed by DeeperLab [207]. Another approach entitled

PanopticDepth [208] introduced a unified framework designed for depth-aware panoptic segmentation (DPS), a complex task with scene understanding. DPS aims to reconstruct a 3D scene with instance-level semantic understanding from a single image, assigning each pixel a depth value, a semantic class label, and an instance ID. Unlike conventional approaches that add a dense depth regression head to panoptic segmentation networks as an independent branch, PanopticDepth employs a dynamic convolution technique to predict instance-specific depth and segmentation masks. The methodology highlights the advantage of the mutually beneficial relations between panoptic segmentation and depth estimation. While the paper primarily focuses on DPS, the model has also the ability to estimate individually panoptic segmentation and depth, offering flexibility in its application. The joint learning of panoptic segmentation and depth suggests that the knowledge gained in one task enhances the performance of the other.

Some alternative cooperative techniques for panoptic segmentation have been proposed [209; 210; 211]. These techniques are also based on a shared backbone architecture in addition to explicit connections between the instance and semantic segmentation heads. The approach outlined in [209] involves using a ShuffleNet [212] for feature extraction, as well as establishing explicit connections between the instance and semantic segmentation stages. These steps are followed by combining the results to produce the final panoptic output. A deep panoptic segmentation method that relies on a bidirectional learning technique is presented in [211]. To capture the intrinsic interaction between semantic and instance segmentation, the authors introduce a Bidirectional Aggregation Network called BANet [211]. This network performs panoptic segmentation by leveraging two modules that extract rich contextual features from semantic and instance segmentation for recognition and localization. Finally, the bidirectional paths are used for feature aggregation, enhancing the overall segmentation performance. On the other hand, the architecture proposed in [210] allows information exchange between the branches to take advantage of both. Specifically, it involves leveraging semantic information to improve the instance segmentation. The output of the semantic segmentation branch is normalized and concatenated with the normalized features of the feature map. This concatenated information is passed through a convolutional layer and used as input to the instance segmentation branch. This allows relevant data from one branch to flow through the other, improving the performance of both semantic and instance segmentation branches.

The first category with a shared backbone and no explicit connections offers several advantages [197; 204; 199]. It provides a straightforward implementation compared to the second category. Additionally, removing explicit connections between the two heads reduces computational complexity during training. The separate heads offer flexibility in optimizing each task independently, allowing for more control over the model behavior. Despite these advantages, this category has also its limitations. The lack of explicit interaction between the heads can result in potential misalignment between the semantic and

instance segmentation tasks. Furthermore, the absence of explicit connections limits the direct exchange of information between the two heads, which may affect the model ability to leverage fine-grained semantic information for accurate instance segmentation or vice versa. On the other hand, the second category also has its advantages [209; 210; 211]. First, the explicit connections enable better integration and information exchange between the tasks. The interaction allows for contextual refinement, where the predictions from one task can help refine the predictions of the other, leading to more accurate and coherent results. However, it is important to note that this architecture comes with increased complexity. The design and implementation of explicit connections are more challenging, and there may be computational overhead during training and inference due to information exchange between the heads. Moreover, the increased interaction between the heads may introduce the risk of over-fitting, as the model can excessively rely on the information exchange and lose generalization capability.

Based on the introduced papers and contributions, it is difficult to definitively conclude that one architecture always outperforms the other in all aspects considering panoptic segmentation. The choice depends on various factors such as the specific DNN architecture, the characteristics of the dataset, etc. Different datasets, tasks, and contexts may favor one architecture over the other. Ultimately the selection should be based on a careful consideration of the trade-offs between simplicity, computational efficiency, integration, and performance, as well as the available resources for training and inference.

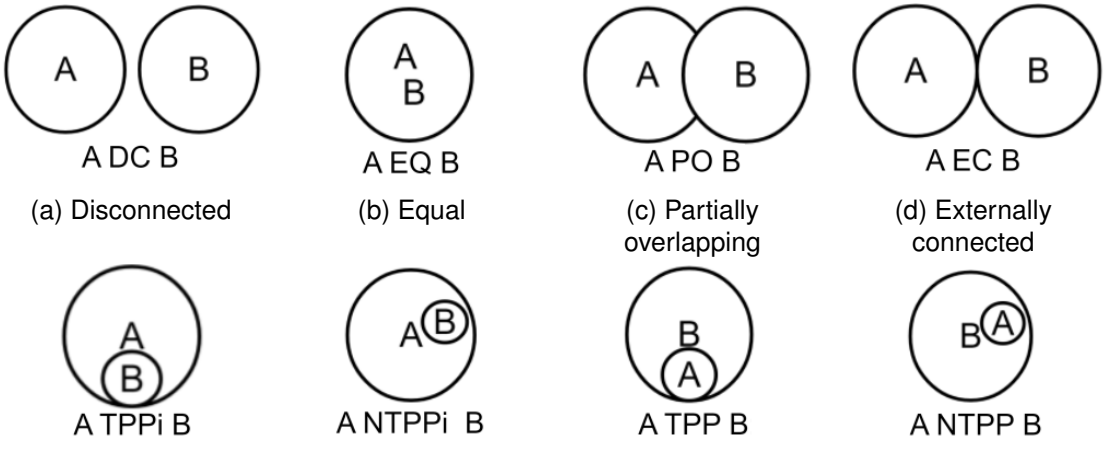
Additionally, from the state-of-the-art chapter (Section 2), it has been demonstrated that the collaborations between knowledge and DL models lead to improved performances and results. By making use of external knowledge sources, DL models can benefit from additional contextual cues, enhancing their capabilities in various computer vision tasks. These integrations have shown promising outcomes, proving the potential of combining knowledge-driven approaches with DL techniques. Building upon these insights, we propose in this chapter to integrate spatial relationship knowledge between objects in urban scenes into DL models dedicated to panoptic segmentation. By incorporating this extra knowledge, we aim to take advantage of the spatial and contextual relationships between objects in urban scenes, which can provide valuable cues for accurate segmentation and scene understanding in complex urban environments. Traditional DL models may struggle to capture the complex spatial arrangements and semantic associations between objects, leading to inefficient segmentation results. However, by explicitly incorporating spatial relationship knowledge, we can enhance the models ability to perceive the overall scene and capture the underlying structure. We propose to deeply incorporate spatial relationships directly into the DNNs training process through the loss function to further advance knowledge integration. This enables the DNN to learn both from the visual data and the extra knowledge simultaneously, leading to a more comprehensive and effective learning process. By jointly optimizing the segmentation task with the spatial re-

lationship knowledge, the model can benefit from the contextual information and refine its predictions consequently. This integrated approach not only enhances the segmentation accuracy but also advances a deeper understanding of the urban scene by considering the relationships between objects.

4.3/ Qualitative Spatial Relationships (QSRs)

The 3D objects of an urban scene are projected into acquired 2D images as geometric regions of different shapes, visual aspects, and sizes. To integrate knowledge representing spatial relationships between these objects, we refer to Qualitative Spatial Relationships (QSRs) [213]. Our approach involves extracting all spatial relationships that exist between every pair of regions within an image and integrating this information into the training process of a DNN as extra knowledge. This integration of complementary relations is expected to enhance the model ability to better understand the spatial structure of the urban environment objects and improve the accuracy of panoptic segmentation prediction results.

Specifically, we are interested in Region Connection Calculus (RCC) [214], which is a standardized set of spatial relations that is used to capture the possible connections and arrangements between regions, allowing for a comprehensive representation of their spatial interactions. There are many versions of these Region Connection Calculus such as RCC-5 and RCC-8. In our case, we considered RCC-8 which describes 8 fundamental relations (Figure 4.3). It offers a fine level of detail that enables a precise representation of the relationships between two regions in an image. Consequently, it enables a more comprehensive spatial understanding of the environment.



(a) Disconnected (b) Equal (c) Partially overlapping (d) Externally connected

(e) Tangential Proper Part inverse (f) Non-Tangential Proper Part inverse (g) Tangential Proper Part (h) Non-Tangential Proper Part

Figure 4.3: Representation of RCC-8 relations

RCC-8 specifically defines eight distinct relationships. Let U denote the set of non-empty regular closed sets, also known as regions. Within the RCC-8 algebra, 8 topological relations serve as its foundation [215]. The Disconnected (DC) relationship (Figure 4.3a) signifies that two regions do not have shared points or boundaries. The Externally Connected (EC) relationship (Figure 4.3d) denotes one region surrounding or enclosing another. The Tangential Proper Part (TPP) relationship (Figure 4.3g) implies that one region is entirely contained within another, with at least one shared boundary point. On the other hand, the Non-Tangential Proper Part ($NTPP$) relationship (Figure 4.3h) indicates complete containment without shared boundaries. The Partially Overlapping (PO) relationship (Figure 4.3c) suggests that the regions have some common points or boundaries, without one region entirely encompassing the other. When both regions are identical in shape and size, they are considered Equal (EQ) (Figure 4.3b). Finally, the Tangential Proper Part Inverse ($TPPi$) (Figure 4.3e) and Non-Tangential Proper Part Inverse ($NTPPi$) (Figure 4.3f) relationships mirror their respective counterparts but with the roles of the regions reversed.

Consider the following examples of common possible and impossible spatial relationships between objects in an urban environment. For example, regions representing a building in an image are typically partially overlapping (Figure 4.3c) with regions corresponding to vegetation class. We can also say that regions representing the sky and buildings are usually externally connected (Figure 4.3b). This relationship means that the sky region surrounds or encloses the building region, as buildings are typically taller structures that are externally connected to the sky. Furthermore, the road and sidewalk regions are often externally connected, indicating their spatial relationship (Figure 4.3d). This relationship means that the sidewalk region is adjacent to and connected to the road region. Moreover, the sidewalk region is always disconnected from the sky region (Figure 4.3a). It means that there are no shared points or boundaries between the sidewalk and the sky. The sidewalk, which is at ground level, is a horizontal surface that is separate from the overhead expanse of the sky. Additionally, a region representing a pedestrian or a car cannot be fully included (Figure 4.3g) within a region of the sky. This is because the sky and the pedestrian or car regions have distinct spatial characteristics and occupy different areas in the scene. A truck region cannot be fully included within a car region (Figure 4.3g). While there may be areas where the truck and car regions partially overlap (Figure 4.3c), the complete inclusion of a truck region within a car region is unlikely due to their different dimensions.

In conclusion, the eight relations we have presented provide a comprehensive and detailed representation of spatial relationships between objects in the urban environment. These relations serve as a formal logic that captures essential spatial knowledge of the components within the environment. By combining this knowledge with the performances of a DNN, we can create an informed DL framework to enhance the network understand-

ing and reasoning abilities. In the next section, we describe the methodology to extract the RCC-8 relations and integrate them into a DNN.

4.4/ Spatial relationships integration for panoptic segmentation

This section presents the proposed DNN architecture that integrates RCC-8 relations between objects perceived in images. It is important to mention that the proposed approach is general and can be applied to any two-head (one for semantic segmentation and the other for instance segmentation) panoptic segmentation model.

As mentioned previously, the main idea of the proposed technique is to optimize and enhance the performance of panoptic segmentation models by incorporating additional knowledge on the spatial relationships between different objects in an urban scene directly during the model training. We aim to integrate this knowledge by introducing a novel loss function that captures and represents the spatial relationships between objects. By incorporating this loss function into the training process, the model gains a comprehensive understanding of the urban environment, improving its ability to accurately segment objects by considering their contextual interactions. To extract the RCC-8 relations between the various object types of the image, including both “Stuff" and “Things", we integrated the proposed module in both heads during the training of the DNN (Figure 4.4). This module is designed to extract the RCC-8 relations between regions to define and compute the proposed $L_{RCC-pano}$ loss function. To do so, distinct image regions should be separated, and then the different regions should be approximated before extracting the RCC-8 relations. An example of the overall methodology to extract the 8 RCC relationships between “Stuff" regions is presented in Figure 4.5.

Separation of distinct regions The proposed module takes as input the “Stuff" regions from the predicted semantic segmentation map and those from the ground truth (Figure 4.5). In the semantic map, “Stuff" regions belonging to the same class are labeled with a common label, even though they are not connected. For example, in Figure 4.5, the two separate regions belonging to the class “Vegetation" were both labeled with the same label (V), despite being distinct and not connected. However, it is important in our case to consider each region independently of the others to accurately represent and integrate the spatial relationships between all the distinct regions in the scene. To solve this problem, we implemented an algorithm that separates all the distinct visible “Stuff" regions from the semantic maps. We also added some identifiers to reference the distinct regions belonging to the same label in both the prediction and the ground truth

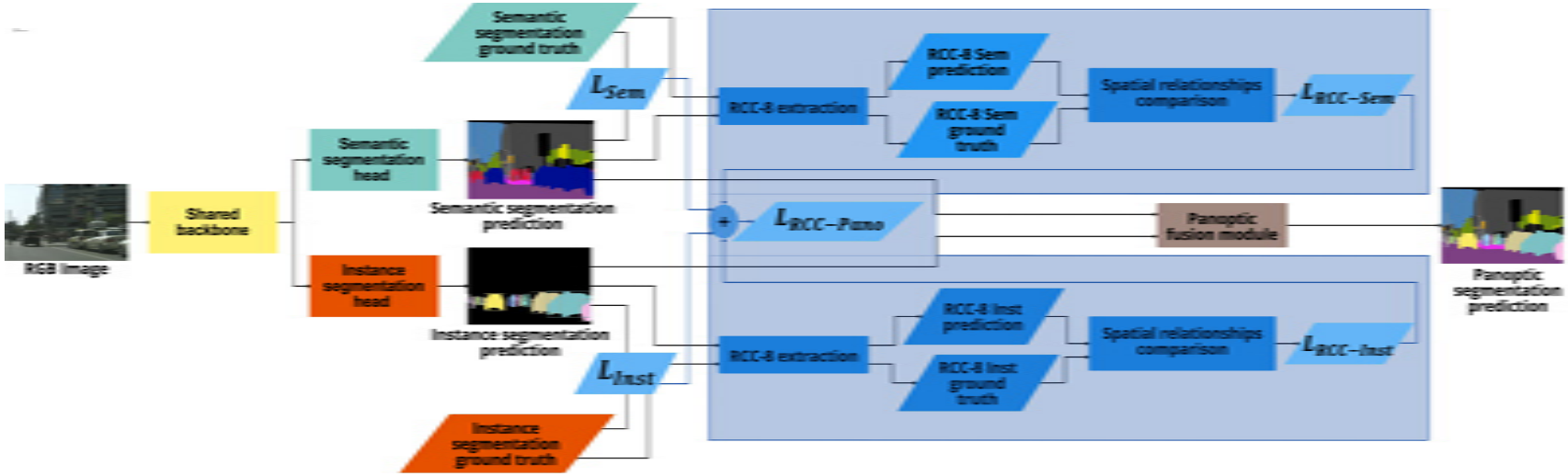


Figure 4.4: The proposed architecture for the integration of spatial relationships into a two-head panoptic segmentation deep neural network. The blue module is our contribution.

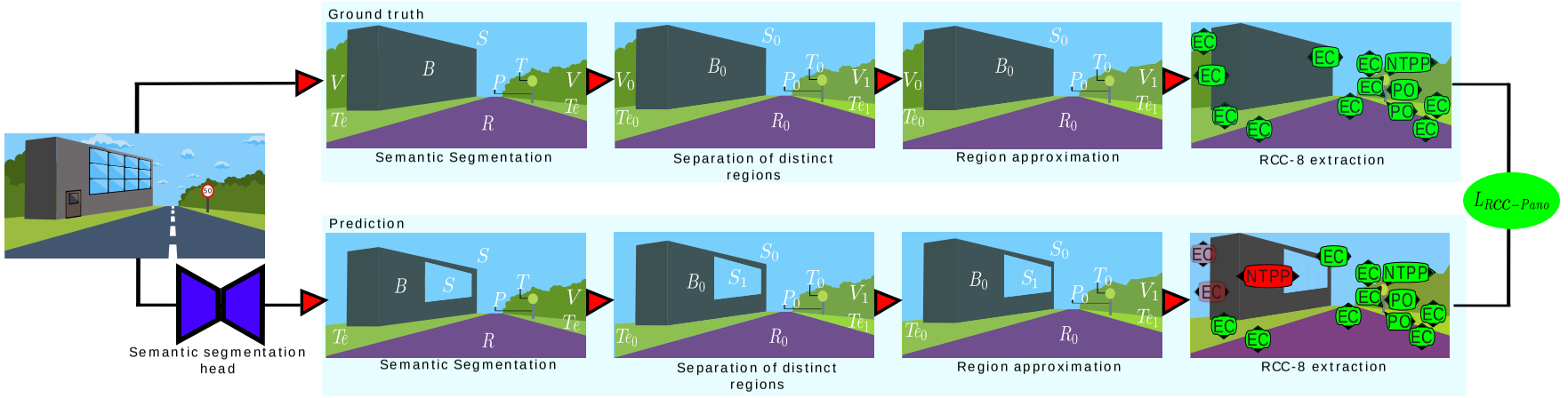


Figure 4.5: Methodology to extract the 8 RCC relationships between “Stuff" regions. The upper block presents the process considering the semantic segmentation ground truth, and the bottom block represents the process for prediction. In the final step, the green relationships indicate correct matches between the ground truth and prediction, the red ones represent false positives, and the red transparent ones represent false negatives.

(Figure 4.5: Separation of distinct regions). Since the concept of instance segmentation itself involves identifying and separating individual objects within an image, we did not face the problem of identifying distinct regions regarding the “Things" regions related to the instance segmentation branch. Thus, each region belonging to an instance is basically segmented separately from the other instances of the same class. At the end of this step, we consider a set of distinct regions for each of the predicted maps (semantic and instance segmentation), along with their respective ground truths regions.

Region approximation To identify the spatial relationships between regions, we initially extracted the primary features and characteristics of each region. Specifically, the centroid coordinates and their principal and secondary axes are computed, which are used to generate a polygon approximation with a maximum of 50 vertices for each region (Figure 4.5: Region approximation). The polygons are used to establish the spatial relationships between each pair of regions. Once again, this technique is applied to both the predicted semantic and instance segmentation maps, as well as their corresponding ground truth regions.

RCC-8 extraction The computed regions properties are used to extract the RCC-8 relations (Figure 4.5: RCC-8 extraction). The goal is to introduce a new penalty term to the global loss function of the panoptic segmentation DNN by comparing the 8 RCC spatial relations in the semantic and instance segmentation prediction maps with their corresponding ground truths. To incorporate these comparative elements into the network training, we propose the addition of two new penalty terms to the loss function, namely L_{RCC-S} and L_{RCC-I} which respectively correspond to the semantic and instance segmentation heads (Figure 4.4). These penalty terms aim to penalize the network errors made among the 8 RCC relations between the image regions during training. Mathematically, L_{RCC-S} and L_{RCC-I} represent the average of the 8 penalty terms of the 8 RCC relations (Figure 4.3):

$$L_{RCC-Sem} = \frac{1}{8}(L_{PO-S} + L_{EC-S} + L_{TPP-S} + L_{NTTP-S} + L_{DC-S} + L_{EQ-S} + L_{TPPi-S} + L_{NTTPi-S}). \quad (4.1)$$

$$L_{RCC-Inst} = \frac{1}{8}(L_{PO-I} + L_{EC-I} + L_{TPP-I} + L_{NTTP-I} + L_{DC-I} + L_{EQ-I} + L_{TPPi-I} + L_{NTTPi-I}). \quad (4.2)$$

L_{RCC-S} and L_{RCC-I} range between 0 and 1 and represent the ability of the neural network to verify the 8 RCC relationships between objects in images. The penalty terms corre-

sponding to the 8 RCC relations are defined as the ratio between the errors made by the model in the corresponding RCC relation and the sum of the wrong and the correct matches of the same relation with the ground truth. For example, if we consider the RCC relation “PO" (Partially Overlapping), the penalty term is defined as follow:

$$L_{PO} = \frac{Errors_{PO}}{Errors_{PO} + Correct_{PO}}. \tag{4.3}$$

To provide a clear illustration, consider the example provided in Figure 4.5. We have an image with its corresponding semantic segmentation ground truth, which contains the following pairwise object “EC" relations: (B_0, V_0) and (V_0, S_0). On the other hand, the semantic segmentation map prediction of the same image does not include these relations and instead, it contains the pairwise object “NTTP" relation: (S_1, B_0). From the comparison, we can identify two types of errors made by the model. The first error is the presence of the “NTTP" relation for the pairwise (S_1, B_0), which does not exist in the ground truth. This can be considered as a false positive since the model incorrectly identified a relationship between the region S_1 and the region B_0. The second error is the failure to detect the (B_0, V_0) and (V_0, S_0) relations, where the model did not recognize the “EC" connections between each pair of regions. These errors can be seen as a false negative since the model missed a true relation that should have been identified. Following the same methodology, all penalty terms for the 8 RCC relations are computed.

In general, DL models for panoptic segmentation that follow an architecture with two heads -one for semantic segmentation and the other for instance segmentation- typically employ a global loss function. The global loss function for these models is commonly defined as the sum of two individual loss functions: L_{Sem}, which optimizes the semantic segmentation head, and L_{Inst}, which optimizes the instance segmentation branch (Figure 4.4). Therefore, the general form of the loss function for such models can be expressed as:

$$L_{Pano} = L_{Sem} + L_{Inst}. \tag{4.4}$$

Using the proposed penalty terms, the new global loss function for optimizing the whole network while considering the integration of the spatial relationships knowledge between the objects is defined as follows :

$$L_{RCC-Pano} = L_{Sem} + L_{Inst} + L_{RCC-Sem} + L_{RCC-Inst}. \tag{4.5}$$

4.5/ EXPERIMENTS AND RESULTS

To validate, evaluate, and demonstrate the performance of integrating spatial relationships knowledge into a DNN for panoptic segmentation, we consider a state-of-the-art panoptic segmentation network (EfficientPS [197]) as our base network. EfficientPS is a robust model that demonstrates exceptional performance in panoptic segmentation compared to other state-of-the-art approaches. It is also highly extensible, making it suitable for making modifications and adding modules to implement the proposed approach. The architecture of the model is presented in Section 4.5.1. The implementation details are described in Section 4.5.2. The panoptic segmentation evaluation metrics are introduced in 4.5.3. In Section 4.5.4, we present the different considered datasets for training and evaluation. The analysis of the quantitative results and the comparison with the state of the art considering the three datasets for urban environments are respectively highlighted in Section 4.5.6, 4.5.7 and 4.5.8. Qualitative results are presented in Section 4.5.9. In Section 4.5.10 we highlight an evaluation on unseen datasets. Ablation study is presented in Section 4.5.11. Generalization capability of the proposed approach and the quantitative analysis of the integration interest of spatial relationships into the learning process are given in Section 4.5.12 and Section 4.5.13 respectively.

4.5.1/ ARCHITECTURE OF THE EFFICIENTPS MODEL

The EfficientPS architecture [197] includes a shared backbone with a 2-way FPN. The shared backbone is based on the EfficientNet architecture [148], which uses mobile inverted bottleneck units [216] and compound scaling to enhance its representational capacity with fewer parameters compared to other similar networks. Instead of using the conventional FPN as most of the state-of-the-art works [180; 217], EfficientPS incorporates a 2-way FPN that effectively fuses multi-scale features in both directions. This is achieved by spreading information flow in multiple directions. After the 2-way FPN, two heads work in parallel: the semantic segmentation head and the instance segmentation head. The instance head is based on a variant of the Mask R-CNN architecture [82], while the semantic segmentation is based on three modules dedicated to the capture of fine features, long-range contextual features, and correlating distinct features for improved object boundary refinement. To produce the panoptic segmentation output, EfficientPS employs a panoptic fusion module that combines the outputs from the semantic and instance heads. This module integrates the predictions from both heads to yield the final panoptic segmentation result. It combines predictions from the semantic and instance segmentation heads to create the panoptic segmentation output. It first considers object instances from the instance segmentation head, then reduces their number based on confidence scores and handles overlapping instances. The masks for each object

instance are combined with the “stuff" to produce intermediate panoptic prediction, from which the final panoptic segmentation output is produced.

4.5.2/ Implementation details

Regarding the implementation of the algorithm for the extraction of the RCC-8 spatial relationships between objects (Section 4.3), we used the Measure Region Properties module of the Scikit-image library [218]. Additionally, we considered the QSRLIB Library [219] to infer the RCC-8 spatial relationships.

The official implementation code is available online. The EfficientPS model [197] is implemented using PyTorch 1.7 Neural Network Libraries with CUDA GPU Toolkit 11.2. The hyper parameters set by the authors have remained unchanged. However, on the EfficientPS paper [197], the training was performed on 16 NVIDIA Titan X 12GB GPUs. The batch size was set to 1 and the number of epochs to 160. Due to our less powerful GPU resources available (2 NVIDIA GeForce RTX 2080 Ti 11GB GPUs), we were unable to train the model under the same conditions. To address this technical challenge, we chose to use the “EfficientNet-b4" as the shared backbone instead of the “EfficientNet-b5" used in [197]. Indeed, the b4 version is lighter than the b5 version, allowing us to train the model based on available computational resources. Table 4.1 presents a complexity comparison between the two Efficient-Net versions.

Encoder	Parameters (M)	FLOPs (B)
EfficientNet-b5	30	250.97
EfficientNet-b4	19	156.49

Table 4.1: Complexity comparison among different versions of the Efficient-Net backbone.

4.5.3/ Evaluation metrics

We use the standard Panoptic Quality metrics of the state of the art [180] to evaluate the performance of the proposed approach. These metrics are presented below.

The Panoptic Quality (PQ) metric quantifies the accuracy of object instance segmentation as well as the correct prediction of the “Stuff" class. It is calculated as follows:

$$PQ = \frac{\sum_{(p,g)\in TP}(IOU(p,g))}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}. \tag{4.6}$$

where $\sum_{(p,g)\in TP}$ represents the sum over all pairs of prediction and ground truth objects that belong to the set TP, which represents the True Positives. FP, and FN, respectively, represent False Positives and False Negatives. IOU denotes the Intersection Over Union

(IOU) ratio, defined as:

$$IOU = \frac{TP}{TP + FP + FN}. \tag{4.7}$$

The Segmentation Quality (SQ) metric indicates the accuracy of the predicted segments in comparison to the ground truth. It is calculated by averaging the IOU scores of all the TP segments. A higher SQ value or a value closer to 1 indicates that TP segments closely align with their corresponding ground-truth segments, while a lower value means poor matching. The SQ metric is defined as:

$$SQ = \frac{\sum_{(p,g)\in TP} IOU(p,g)}{|TP|}. \tag{4.8}$$

However, SQ focuses on evaluating the accuracy of TP, without considering the FN nor FP segments. To consider the impact of incorrect predictions, the Recognition Quality (RQ) is introduced as a metric that combines precision and recall. RQ aims to provide a comprehensive assessment of the model effectiveness in correctly identifying objects in the image, considering both the ability to avoid false positives and false negatives. The RQ metric is defined as:

$$RQ = \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}. \tag{4.9}$$

Following the standard benchmarking criteria for panoptic segmentation, we calculate PQ, SQ, and RQ for all the dataset classes, and also report them separately for “Stuff" classes (PQ_{st},SQ_{st} and RQ_{st}) and “Things" classes (PQ_{th},SQ_{th} and RQ_{th}).

4.5.4/ Datasets

In this section, we introduce the considered datasets for our experiments. We chose to use a range of challenging urban datasets to validate and demonstrate the effectiveness of the proposed approach across multiple scenarios and different conditions.

CityScapes The CityScapes dataset [154] is a challenging dataset for panoptic segmentation, as it consists of diverse urban street scenes from more than 50 European cities, captured under different conditions. The scenes are crowded with dynamic objects such as pedestrians and cars that are mostly grouped and occluded, making the panoptic segmentation task challenging. Recently, the CityScapes dataset introduced a benchmark for panoptic segmentation, with pixel-level annotations for 19 object classes, including 11 “Stuff" classes and 8 “Things" classes. The dataset contains 5000 finely

annotated images captured using a stereo camera with a resolution of 2048×1024 pixels. These images are divided into 2975 images for training, 500 images for validation, and 1525 images for testing. However, the annotations for the test set are not publicly available. Following the state-of-the-art protocol, we evaluated our model on the validation set.

KITTI The KITTI panoptic segmentation dataset for urban scene understanding was introduced by the authors of [197]. It includes panoptic annotations for a subset of 1055 images from the KITTI Vision Benchmark Suite [156]. These images are divided into 855 images for training and 200 images for validation. The dataset provides annotations for 11 “Stuff" classes and 8 “Things" classes, following the distribution of classes in the CityScapes dataset [154].

Indian Driving Dataset The Indian Drivind Dataset (IDD) [220] addresses the scene understanding challenge of unstructured environments. This dataset contains more “Things" classes per scene compared to other datasets such as KITTI or CityScapes. However, the categories for traffic objects are somehow limited. The images of this dataset were captured using a front-facing camera, gathering data from two Indian cities. IDD includes a total of 10003 images with 6993 images for training, 981 for validation, and 2029 for testing. The images have a resolution of either 1920×1080 pixels or 720×1280 pixels.

4.5.5/ Training protocol

As indicated in the training protocol of the original EfficientPS [197], we initialized the backbone of the proposed model with weights pre-trained on the ImageNet dataset [20]. For training, we used Stochastic Gradient Descent (SGD) with a momentum of 0.9 and employed a multistep learning rate schedule. We started with an initial base learning rate of 10^{-4}, then, at each milestone, we reduced the learning rate by a factor of 10 and continued training until convergence. We trained the proposed model with a batch size of 2 on 2 NVIDIA GeForce RTX 2080Ti 11GB GPUs.

4.5.6/ Evaluation on CityScapes Dataset

In this section, we present a comparative analysis of the proposed approach against current state-of-the-art panoptic segmentation methods. We evaluated and compared our technique on the CityScapes dataset [154] and report the performance metrics in Table

4.2 and Table 4.3 with the results mentioned in the corresponding papers of the state-of-the-art methods. For a complete evaluation, we evaluated both the CityScapes dataset validation and test sets. However, as mentioned in Section 4.5.2, we were unable to train the EfficientPS model [197] with its original configuration due to the limitations of our computational resources. Instead, we used the officially released implementation provided by the authors of EfficientPS and retrained the model according to our resources. Specifically, we retrained the model using the “EfficientNet-b4" backbone, which is a lighter version compared to “EfficientNet-b5" (see Table 4.1).

Evaluation on CityScapes validation set To build a comprehensive benchmark for our model performance, we first performed our evaluation on the CityScapes validation set. We made this choice because the CityScapes validation set is publicly available and widely adopted by most state-of-the-art methods as a common reference for evaluation. By focusing on this subset, we can compare it with other approaches, allowing us to position our method among the existing ones. In Table 4.2, the baseline approach “EfficientPS-b4" yields a PQ of 60.6, an SQ of 80.3, and an RQ of 74.3, with a PQ(th) and a PQ(st) of 56.3 and 63.8 respectively. However, the proposed approach, which incorporated additional knowledge about spatial relationships between objects in the loss function during model training, achieved higher scores. Specifically, it provided a PQ of 64.2, an SQ of 81.6, and an RQ of 77.5. The PQ(th) and the PQ(st) also respectively improved to 59.8 and 67.6. Furthermore, in comparison with prior state-of-the-art works, the proposed approach demonstrates superior performances regarding the panoptic evaluation metrics. These results highlight the effectiveness of integrating spatial relationships into the panoptic segmentation neural network. The improved PQ, SQ, and RQ scores signify that the proposed approach outperforms the baseline in terms of overall panoptic, segmentation, and recognition quality. More specifically, the improved RQ score indicates an enhanced recognition quality, suggesting that the proposed approach is better at accurately identifying and classifying objects in the scene. This means that the model developed a higher ability to recognize and assign correct labels to instances and semantic classes within the image thanks to the integrated RCC knowledge. Similarly, the higher SQ score indicates improved segmentation quality. This suggests that the proposed approach achieves more precise and accurate object boundaries, resulting in a better overall representation of the scene. Another observation is that the proposed approach “EfficientPS-b4-RCC" not only outperforms the baseline implementation “EfficientPS-b4" but also achieves higher performance compared to the original architecture of the paper with a heavy backbone “EfficientPS-b5". This demonstrates that we can achieve high performance without being reliant on a powerful computational infrastructure. Instead, we can leverage other methods, such as incorporating meaningful knowledge, to enhance the network performance. Despite the overall out-performance of the proposed approach

Method	PQ	SQ	RQ	PQ(th)	SQ(th)	RQ(th)	PQ(st)	SQ(st)	RQ(st)
WeaklySupervised [221]	47.3	–	–	39.6	–	–	52.9	–	–
DeeperLab [207]	56.3	–	–	–	–	–	–	–	–
Panoptic FPN [222]	58.1	–	–	52.0	–	–	62.5	–	–
AUNet [223]	59.0	–	–	54.8	–	–	62.1	–	–
UPSNet [224]	59.3	79.7	73.0	54.6	79.3	68.7	62.7	80.1	76.2
Seamless [217]	60.3	–	–	56.1	–	–	63.3	–	–
SSAP [225]	61.1	–	–	55.0	–	–	–	–	–
AdapTIS [226]	62.0	–	–	58.7	–	–	64.4	–	–
Panoptic-DeepLab [198]	63.0	–	–	–	–	–	–	–	–
EvPSNet [227]	63.7	81.3	77.5	–	–	–	–	–	–
EfficientPS-b5 [197]	63.9	81.5	77.1	60.7	81.2	74.1	66.2	81.8	79.2
PanopticDepth [208]	64.1	–	–	58.8	–	–	**68.1**	–	–
EfficientPS-b4	60.6	80.3	74.3	56.3	79.2	70.9	63.8	81.1	76.7
EfficientPS-b4-RCC	**64.2**	**81.6**	**77.5**	**59.8**	**80.3**	**73.8**	**67.6**	**82.4**	**80.2**

Table 4.2: Comparison of panoptic segmentation performance on the CityScapes validation set. (st) and (th), respectively, denote the “Stuff" and “Things" classes. “–" indicates the unreported metric for the corresponding method.

that reaches respectively a PQ, an SQ, and an RQ of 64.2, 81.6, and 77.5 compared to the “EfficientPS-b5" with a PQ of 63.9, SQ of 81.5 and a RQ 77.1, it is important to analyze the specific challenges associated to the Things-based metrics, where the proposed approach is somehow similar to the original one. Our approach reaches a PQ(th), an SQ(th), and an RQ(th) respectively of 59.8, 80.3, and 73.8 while “EfficientPS-b5" provided a PQ(th) of 60.7, an SQ(th) of 81.2 and an RQ(th) of 74.1. However, considering “Stuff" objects, our approach reaches a PQ(st), an SQ(st) and an RQ(st) respectively of 67.6, 82.4 and 80.2 while “EfficientPS-b5" provided a PQ(st) of 66.2, an SQ(st) of 81.8 and an RQ(st) of 79.2. One possible explanation could be the nature of the scene composition itself. In the majority of urban scenes, the pixels related to “Stuff" objects, such as sky, vegetation, and road are numerous than the “Things" objects which makes it more difficult to reach an improvement considering the instance-level objects. Furthermore, the proposed approach focuses on integrating knowledge about spatial relationships between objects in the loss function during model training. This integration enables the model to learn the overall composition of the scene, which could be particularly advantageous for segmenting “Stuff" objects. They often have distinct boundaries and homogeneous regions, making them more receptive to contextual cues and spatial relationships. For example, knowing that the sky usually appears above vegetation or buildings helps the model refine the segmentation boundaries and produces more accurate results for “Stuff" objects. However, when it comes to instance-level object segmentation, the situation is more complex. For example, it is challenging to establish a fixed spatial relationship between pedestrians and cars, as pedestrians can be found in various locations, move dynamically, and have diverse interactions with their surroundings.

Evaluation on CityScapes test set After evaluating our approach on the Cityscapes validation set and observing promising results, we proceeded to extend our analysis to the Cityscapes test set (Table 4.3). Similarly to our findings on the validation set, we observed that our proposed approach, “EfficientPS-b4-RCC", consistently outperformed the baseline, “EfficientPS-b4", and also the original architecture, “EfficientPS-b5" which employs a heavier backbone. This suggests that our approach can deliver high performance without the need for extensive computational resources, highlighting the efficacy of incorporating meaningful knowledge to enhance network performance. In terms of panoptic segmentation metrics, our approach maintained its superior performance, achieving a PQ, an SQ, and an RQ of 64.5%, 83.0%, and 77.30% respectively, compared to “EfficientPS-b5" with a PQ of 64.1%, an SQ of 82.6%, and an RQ of 76.8%. To conclude, the integration of spatial relationships in the loss function during the training of the EfficientPS model likely facilitated the model ability to capture contextual information, mainly the spatial layout of scene objects, which enhanced its panoptic segmentation accuracy. This additional knowledge allowed the model to better understand and use the spatial context of objects in the image, resulting in improved performance in terms of PQ, SQ, and RQ metrics. In addition to the global PQ metric that has been increased thanks to our approach, the RQ and SQ metrics were also improved. This means that incorporating the 8 RCC relationships into the model loss function has also increased the model ability to accurately recognize and distinguish between instances of different objects, leading to higher RQ scores. Furthermore, the models ability to precisely segment objects has significantly enhanced as indicated by the SQ metric.

Method	PQ	SQ	RQ	PQ(th)	SQ(th)	RQ(th)	PQ(st)	SQ(st)	RQ(st)
SSAP [225]	58.9	82.4	70.6	48.4	–	–	66.5	–	–
Unifying [228]	61.0	81.4	73.9	52.7	79.6	66.2	67.1	82.8	79.6
Panoptic-DeepLab [198]	62.3	82.4	74.8	52.1	–	–	69.7	–	–
PanopticDepth [208]	62.0	–	–	55.0	–	–	67.1	–	–
EfficientPS-b5 [197]	64.1	82.6	76.8	**56.7**	–	–	69.4	–	–
EfficientPS-b4	61.1	81.6	73.9	53.3	80.4	66.3	66.9	82.5	79.5
EfficientPS-b4-RCC	**64.5**	**83.0**	**77.3**	56.6	**83.7**	**69.7**	**70.2**	**85.8**	**82.8**

Table 4.3: Comparison of panoptic segmentation performance on the CityScapes test set. (st) and (th), respectively, denote “Stuff" and “Things" classes. “–" indicates unreported metric for the corresponding method.

Evaluation of the model complexity In section 4.5.2, we have presented the complexity of different backbone architectures (Table 4.1). In Table 4.4, we present a comprehensive comparison of the entire proposed model “EfficientPS-b4-RCC" with other state-of-the-art methods and the original EfficientPS architecture employing EfficientNet-b5 as its backbone.

Method	Input Size (pixels)	Parameters (M)	FLOPs (B)	Inference time (ms)
DeeperLab[207]	2049×1025	–	–	463
UPSNet [224]	2048×1024	45.05	487.02	202
Seamless [217]	2048×1024	51.43	514.00	168
Panoptic-DeepLab [198]	2049×1025	46.73	547.49	175
EfficientPS-b5 [197]	2048×1024	40.89	433.94	166
EfficientPS-b4	$\mathbf{2048 \times 1024}$	**29.89**	**339.46**	**159**
EfficientPS-b4-RCC	$\mathbf{2048 \times 1024}$	**29.89**	**339.46**	**159**

Table 4.4: Comparison of the model complexity with state-of-the-art panoptic segmentation architectures. “—" indicates unreported metric for the corresponding method.

The comparison table provides insights into several key metrics, including the number of parameters, FLOPs, and inference time while considering the input data size. We observe that the proposed model, “EfficientPS-b4-RCC", has a lower number of parameters, reduced FLOPs, and better inference time compared to other state-of-the-art approaches. Furthermore, it is essential to highlight that both “EfficientPS-b4" and “EfficientPS-b4-RCC" share identical parameters and inference times. Indeed, the main factor that affects inference time is the model architecture and its computational requirements. Since the architecture of the two models is identical, the forward pass during inference will involve the same operations for both models and therefore the inference times are similar. The only major difference is that EfficientPS-b4-RCC has a different loss during training, which does not impact the inference time.

4.5.7/ Evaluation on the KITTI dataset

In this section, we present a comparative analysis of the proposed approach against current state-of-the-art panoptic segmentation methods trained and evaluated on the KITTI dataset. However, there have been only a few state-of-the-art panoptic segmentation methods trained and evaluated on the KITTI, due to the recent proposal of the dataset for panoptic segmentation task in [197]. To compare our proposed approach against the current state-of-the-art methods, we report the results in table 4.5.

Method	PQ	SQ	RQ	PQ(th)	SQ(th)	RQ(th)	PQ(st)	SQ(st)	RQ(st)
Panoptic FPN [222]	38.6	70.4	51.2	26.1	68.3	40.1	47.6	71.9	59.2
UPSNet [224]	39.1	70.7	51.7	26.6	68.5	40.6	48.3	72.4	59.8
Seamless [217]	41.3	71.7	52.3	28.5	69.2	42.3	50.6	73.6	59.6
EfficientPS-b5 [197]	42.9	72.7	53.6	30.4	69.8	43.7	52.0	74.9	60.9
EfficientPS-b4	38.7	72.8	48.7	29.2	69.2	41.5	45.6	71.8	56.4
EfficientPS-b4-RCC	**43.3**	**74.8**	**53.6**	**30.1**	**69.8**	**43.1**	**52.6**	**75.1**	**61.1**

Table 4.5: Comparison of panoptic segmentation performance on the KITTI validation set. (st) and (th), respectively, denote the “Stuff" and “Things" classes.

We follow the same protocol used in our experiments on the CityScapes dataset (Section 4.5.6). The obtained results with “EfficientPS-b4-RCC", showed a significant improve-

ment in all metrics compared to the base experiment “EfficientPS-b4". Specifically, we observed an improvement in the PQ metric from 38.7% to 43.3%. In addition, we achieved better results in terms of SQ and RQ metrics, providing more evidence about the effectiveness of knowledge integration in enhancing region segmentation and recognition.

When comparing our approach with “EfficientPS-b5", we consistently outperformed the global (PQ, SQ, and RQ) evaluation metrics. These findings further validate our initial observations regarding the impact of knowledge integration on model performance for panoptic segmentation, without the need for complex architectures. Furthermore, our approach successfully outperformed all other state-of-the-art methods (Table 4.5). Even if the “Things" evaluation metrics are almost similar to the original model for the reasons mentioned in Section 4.5.6, our approach still demonstrates various advantages. First, the key comparison lies in evaluating the same model architecture with and without knowledge integration to effectively define the impact of the proposed approach, which always consistently yields valuable improvements. Secondly, while there may be slight variations when analyzing "Stuff" and "Things" separately we observe an important enhancement regarding the global panoptic quality metric (PQ) which is the most important.

4.5.8/ Evaluation on IDD dataset

The evaluation results on the IDD dataset are presented in Table 4.6, and their analysis reveals several observations. First, it is evident that the proposed approach consistently outperforms the base implementation “EfficientPS-b4" and the state-of-the-art approaches. We succeeded in reaching a global PQ of 51.2% considering the proposed approach compared to 48.5% with the base implementation. Additionally, the improvement in the Panoptic Quality metrics indicates the effectiveness of the proposed approach in capturing the complex structures of the IDD dataset, which contains unstructured urban environments and scenes with limited road infrastructure boundaries. These findings confirm the robustness of the proposed approach, as it demonstrates its ability to outperform existing methods and achieve enhanced results even in challenging scenarios.

Method	PQ	SQ	RQ	PQ(th)	SQ(th)	RQ(th)	PQ(st)	SQ(st)	RQ(st)
Panoptic FPN [222]	45.9	75.9	60.8	46.1	77.8	60.9	45.8	74.9	60.7
UPSNet [224]	46.6	76.5	60.9	47.6	78.9	61.1	46.0	75.3	60.8
Seamless [217]	47.7	77.2	61.2	48.9	79.5	61.5	47.1	76.1	61.1
EfficientPS-b5 [197]	50.1	78.4	62.0	50.7	80.6	61.6	49.8	77.1	62.2
EfficientPS-b4	48.5	76.6	61.1	47.8	77.8	61.0	48.9	76.0	61.1
EfficientPS-b4-RCC	**51.2**	**78.9**	**64.4**	**50.2**	**80.3**	**61.6**	**52.1**	**79.8**	**64.8**

Table 4.6: Comparison of panoptic segmentation performance in the IDD validation set. (st) and (th), respectively, denote the “Stuff" and “Things" classes.

4.5.9/ QUALITATIVE RESULTS

To evaluate the effectiveness of the proposed approach, we performed a qualitative comparison between the visual results generated using the original “EfficientPS-b4" and the proposed “EfficientPS-b4-RCC". Specifically, the proposed approach helped to achieve better segmentation of objects in the image and also improved the recognition ability of their corresponding classes.

(a) (b) (c)

Figure 4.6: First and second rows represent qualitative comparisons on the CityScapes validation set. The third and fourth rows represent qualitative comparisons on the KITTI validation set. The fifth and sixth rows represent qualitative comparisons in the IDD validation set. (a) RGB images, (b) the visual results generated with “EfficientPS-b4" and (c) the visual results generated with “EfficientPS-b4-RCC".

In the first image from the CityScapes dataset (Figure 4.6, row 1), our approach successfully detected and segmented the pole region surrounded by two regions of the class car. These pixels belong to a portion of the image that contains multiple regions of different

classes with complex spatial relationships, i.e., the pixels belonging to the pole region are externally connected to the car regions. In contrast, the visual result obtained using the original model failed to accurately segment this region. In the second image, we provided an example of a crowded scene (Figure 4.6, row 2) with numerous pedestrians crossing the road simultaneously. Due to the overlapping instances of pedestrians, accurately segmenting them may be challenging. Upon reviewing the visual results, the original model failed to detect several regions of pedestrians. In contrast, the “EfficientPS-b4-RCC" demonstrated a significant improvement in accurately detecting those regions.

Our findings also indicate a significant visual enhancement considering the proposed approach when evaluating its performance on the KITTI dataset (Figure 4.6). The qualitative results show that our method successfully detects objects and classes that were previously undetectable using the base implementation such as the bicycles and pole regions in the first example (Figure 4.6, row 3). Furthermore, we observed an important improvement in accurately segmenting object boundaries, particularly in complex and dense populated areas. For example, we can observe an improvement in accurately segmenting the traffic light compared to the baseline approach (Figure 4.6, row 4). Additionally, the proposed approach effectively captures the lower part of the pedestrians body, providing more precise segmentation results.

Considering the visual results of the IDD dataset, we can observe important enhancements in accurately detecting pole and truck regions in the first image (Figure 4.6, row 5) when considering the proposed approach. Moreover, in the second image of the IDD dataset (Figure 4.6, row 6), we have a crowded scene where multiple regions were detected correctly with “EfficientPS-b4-RCC" such as motorcycles and cars. The qualitative results presented in this section further validate the findings concluded from the quantitative results (Sections 4.5.6, 4.5.7 and 4.5.8). They provide evidence that integrating spatial relationships in the training of a model through its loss function can significantly improve the performance of the model and the quality of the panoptic segmentation.

4.5.10/ Evaluation on Unseen datasets

The evaluation of a DNN on unseen data is an important step to ensure that the network is robust, can generalize well, and is efficient for real-world applications. In this context, and to further validate our conclusions, we performed additional experiments on unseen datasets. The results are presented in Table 4.7.

Cross-validations were performed to ensure comprehensive testing across various datasets. Specifically, we evaluated the models “EfficientPS-b4" and “EfficientPS-b4-RCC" trained on the CityScapes dataset, the KITTI dataset, as well as the IDD dataset. Additionally, we evaluated the performance of the models trained on the KITTI dataset on

Method	PQ	SQ	RQ	PQ(th)	PQ(st)
Models trained on CityScapes and evaluated on KITTI					
EfficientPS-b4	33.1	70.1	46.3	24.1	41.7
EfficientPS-b4-RCC	**36.2**	**71.6**	**49.9**	**26.8**	**44.3**
Models trained on CityScapes and evaluated on IDD					
EfficientPS-b4	37.6	65.4	51.2	37.5	39.2
EfficientPS-b4-RCC	**39.2**	**68.1**	**52.6**	**39.3**	**41.7**
Models trained on KITTI and evaluated on CityScapes					
EfficientPS-b4	45.1	72.5	57.3	39.8	48.4
EfficientPS-b4-RCC	**49.4**	**77.2**	**61.9**	**44.0**	**53.4**
Models trained on KITTI and evaluated on IDD					
EfficientPS-b4	39.8	67.1	53.0	39.5	41.3
EfficientPS-b4-RCC	**41.8**	**70.5**	**56.4**	**42.1**	**44.6**
Models trained on IDD and evaluated on KITTI					
EfficientPS-b4	32.8	68.5	45.1	23.2	40.1
EfficientPS-b4-RCC	**35.9**	**71.2**	**48.4**	**26.2**	**43.5**
Models trained on IDD and evaluated on CityScapes					
EfficientPS-b4	44.2	70.9	55.6	37.8	46.6
EfficientPS-b4-RCC	**47.1**	**72.9**	**57.5**	**39.1**	**48.9**

Table 4.7: Evaluation of panoptic segmentation performance of the proposed “EfficientPS-b4-RCC" on unseen datasets.

the CityScapes and the IDD. Finally, we evaluated the models trained on the IDD dataset considering the KITTI and the CityScapes datasets.

Considering all the experiments performed on unseen datasets (Table 4.7), we identify better performances when employing the proposed approach compared to the basic implementation. Specifically, there was an average improvement of 3% on the Panoptic Quality metric (PQ) when incorporating spatial relationship knowledge. This improvement is particularly important given the challenges of unseen urban environments. Furthermore, the ability of the “EfficientPS-b4-RCC" model to demonstrate improvement on unseen datasets, where it is evaluated on completely different datasets, proves its capacity to learn and adapt to the general context of urban environments, particularly in terms of spatial object relationships. This indicates that the “efficientPS-b4-RCC" model has developed a deeper understanding of the layering of objects in urban scenes and can successfully apply this knowledge across different environments and datasets. By conducting evaluations on unseen datasets, we validate the effectiveness and applicability of our approach beyond the training domain.

4.5.11/ Ablation Study

In this section, we present an ablation study to investigate the impact of adding additional knowledge about spatial relationships between objects in either the semantic segmentation head or instance segmentation head, or both heads. The improvement of the Panoptic Quality metrics (PQ, SQ, and RQ) when integrating spatial relationships in both

the semantic and instance segmentation heads suggests that the additional information provided is useful and advantageous for the overall panoptic segmentation task. This is mainly because spatial relationships can provide meaningful information and help disambiguate spatial arrangements between objects, especially in crowded and complex scenes. Since the architecture of the considered DNN is based on two separate heads for instance and semantic segmentation with a shared backbone, we conducted additional experiments to identify and analyze the impact of adding the spatial knowledge in each of the two heads.

Based on the results reported in Table 4.8, we observe that the integration of spatial relationships in the semantic segmentation head leads to better results (a PQ of 63.4) compared to the incorporation of the same type of knowledge in the instance segmentation head (a PQ of 62.1). One hypothesis is that the reason why the first experiment “EfficientPS-b4-RCC-Sem" performed better could be that the “Stuff" classes are more numerous and diverse than the “Things" classes, and incorporating spatial relationships can help to better distinguish between them. This is because these regions are usually larger compared to “Things" classes regions and occupy a significant portion of the image. Therefore, improving the panoptic segmentation accuracy of these “Stuff" regions will automatically have a significant impact on the panoptic segmentation overall quality.

Method	PQ	SQ	RQ	PQ(th)	SQ(th)	RQ(th)	PQ(st)	SQ(st)	RQ(st)
EfficientPS-b4	60.6	80.3	74.3	57.0	79.2	70.9	62.8	81.1	76.7
EfficientPS-b4-RCC-Inst	62.1	80.9	75.6	58.6	79.6	72.8	64.8	81.7	78.0
EfficientPS-b4-RCC-Sem	63.4	81.3	76.7	59.3	80.1	73.2	66.7	82.6	79.5
EfficientPS-b4-RCC	**64.2**	**81.6**	**77.5**	**59.8**	**80.8**	**73.8**	**67.6**	**83.1**	**80.2**

Table 4.8: Ablation study: “EfficientPS-b4-RCC-Int" refers to adding spatial relationships knowledge only on the instance segmentation head. “EfficientPS-b4-RCC-Sem" refers to adding spatial relationships knowledge only on the semantic segmentation head. All the experiments are performed considering training and evaluation on the CityScapes dataset.

Moreover, we can say that the smaller improvement in the performance of the instance segmentation head when incorporating spatial relationships could be explained by the fact that the instance segmentation task itself is more focused on capturing precise object boundaries and relationships compared to semantic segmentation. Therefore, the addition of spatial knowledge may provide less significant additional information for instance segmentation. However, the combination of both heads with integration of spatial relationships still leads to an overall improvement in panoptic segmentation (a PQ of 64.2). This suggests that the two heads are complementary and provide different types of information that are both useful for panoptic segmentation. In addition, by incorporating spatial knowledge in both heads, we can say that additional meaningful and important features are provided and transferred to the shared backbone of the model. This certainly has led to better feature representation and extraction for both heads, which contributes to the

overall improvement in panoptic segmentation performance.

4.5.12/ GENERALIZATION CAPABILITY

To ensure the impact of integrating spatial relationships within the loss function on DNN performance regardless of both the model architecture, we expanded and applied our methodology to include two additional state-of-the-art models: Panoptic DeepLab [198] and PanopticDepth [208]. In particular, we considered the evaluation of the abovementioned state-of-the-art methods that are different in terms of architecture and performance when compared to EfficientPS model. The goal is to evaluate these models on the CityScapes dataset, a benchmark for panoptic segmentation in urban driving scenarios. This evaluation aims to demonstrate the effectiveness of our approach across various DNNs architectures.

4.5.12.1/ EVALUATION OF PANOPTIC DEEPLAB ON CITYSCAPES VALIDATION SET

The authors of Panoptic DeepLab [198] introduce a novel approach to panoptic segmentation that combines semantic and instance segmentation into a unified framework. Panoptic DeepLab employs two distinct heads dedicated to semantic and instance segmentation predictions. The predicted semantic segmentation and instance segmentation are fused to generate the final panoptic segmentation result. Further details about the architecture of Panoptic DeepLab are provided in Section 4.2.

Experimental setup We follow the same training protocol as in the original paper [198]. Specifically, we use the "poly" learning rate policy [229] with an initial learning rate set at 0.001. The training process involves fine-tuning the batch-normalization parameters, implementing random scale data augmentation, and optimizing the model using Adam optimizer, excluding weight decay. For the CityScapes dataset, the optimal configuration is achieved by training with whole images (size equal to 2049×1025) and using a batch size of 32. We conducted training iterations for a total of 60k steps. The baseline results are derived from single-scale inference. Integration of spatial relationships and subsequent loss calculation remains consistent with the proposed methodology in Section 4.4.

Results and discussion In our analysis, we initially conducted training on our computational resources using the original Panoptic DeepLab model without any modifications. This preliminary step is consistently performed to establish a baseline specific to our computational environment. The goal behind this approach is to ensure consistency and eliminate potential variations introduced by different environments and machines, as

these factors may lead to slightly different model performances. The results of this experiment are presented in Table 4.9: Panoptic DeepLab-Orig (Our Impl.). These results provided a reference for future comparisons. Following this, we refined and trained our model by incorporating the proposed loss function with the integration of RCC relationships, resulting in Panoptic DeepLab-RCC. In particular, our approach demonstrated an important improvement over baseline, achieving an increase of 2.5% in the general PQ evaluation metric. It also demonstrated higher values for all the stuff and things-related metrics. This enhancement further validates the results obtained from the experiments conducted on the EfficientPS model and proves that the approach is effective regardless of the model architecture.

Method	PQ	SQ	RQ	PQ(th)	SQ(th)	RQ(th)	PQ(st)	SQ(st)	RQ(st)
Panoptic DeepLab-Orig [198]	63.0	–	–	–	–	–	–	–	–
Panoptic DeepLab-Orig (Our Impl.)	62.7	81.3	75.8	59.6	80.0	73.2	65.1	80.7	78.2
Panoptic DeepLab-RCC	**65.2**	**83.7**	**78.1**	**62.1**	**82.4**	**75.7**	**67.7**	**83.2**	**80.5**
PanopticDepth-Orig [208]	64.1	–	–	58.8	–	–	68.1	–	–
PanopticDepth-Orig (Our Impl.)	63.7	82.5	76.6	57.7	81.6	70.4	68.0	83.1	81.1
PanopticDepth-RCC	**65.1**	**83.2**	**77.8**	**60.0**	**83.1**	**72.5**	**68.7**	**83.3**	**81.7**

Table 4.9: Comparison of panoptic segmentation performance of Panoptic DeepLab [198] and PanopticDepth [208] models on the CityScapes validation set. (st) and (th), respectively, denote “Stuff” and “Things” classes. “—” indicates unreported metric for the corresponding method.

4.5.12.2/ Evaluation of Panoptic Depth on CityScapes validation set

The authors of PanopticDepth [208] introduce a unified framework designed for depth-aware panoptic segmentation (DPS), a complex task with scene understanding. DPS aims to reconstruct a 3D scene with instance-level semantic understanding from a single image, assigning each pixel a depth value, a semantic class label, and an instance ID. More details about the architecture of this model are presented in Section 4.2. The model can individually evaluate the panoptic segmentation task with its metrics. In this context, to evaluate our approach on the PanopticDepth model, we consider comparing the performance of our method with the results of this model on the CityScapes dataset for the panoptic segmentation task. This is particularly interesting for our approach. Indeed, if the proposed methodology, consisting of the integration of knowledge related to spatial relationships, succeeds in improving performance, it means that even when the model is helped with additional information beneficial to the task, such as depth in this case, the knowledge related to RCC-8, represented as a loss function, remains valuable and

useful.

Experimental setup We follow the same training protocol as in the original paper [208]. The training process of the panoptic segmentation model (PanopticFCN [230]) is divided into two distinct steps. In the initial step, a large mini-batch of small cropped images is employed. During this phase, the model is trained with the Adam optimizer [231] for $130k$ iterations, with synchronized batch normalization. The learning rate is initialized at 0.0001, and a poly schedule with a power of 0.9 is adopted. Images are resized with random factors within the range of $[0.5, 2.0]$, followed by cropping to the 1024×512 dimension. Each mini-batch contains 32 samples. Color augmentation and horizontal flipping are applied during training. In the second phase, the panoptic segmentation model is fine-tuned with images scaled by $[1.0, 1.5]$ and cropped into dimensions of 2048×1024 for additional $10k$ iterations. The batch size is reduced to 8. The integration of spatial relationships and the subsequent calculation of loss remain consistent with the methodology proposed in Section 4.4.

Results and discussion PanopticDepth-Orig (Our Impl.), as reported in Table 4.9, initially achieved a PQ of 63.7% with a PQ(st) of 68.0% and a PQ(th) of 57.7%. Training the model with the proposed approach (Panoptic Depth-RCC) demonstrates improvement across all metrics, including things and stuff-related metrics. This suggests a significant enhancement in the model ability to detect and classify the objects regardless of their type thanks to the proposed methodology. Specifically, the proposed integration of the RCC loss function contributes to this improvement, highlighting the importance of spatial relationships in panoptic segmentation.

To conclude, the experiments conducted in Sections 4.5.12.1 and 4.5.12.2 validate the generalization capability of our approach, showcasing its effectiveness in improving panoptic segmentation models regardless of their architectures. Whether applied to EfficientPS [197], Panoptic DeepLab[198], or PanopticDepth [208], our approach consistently yields improved results. Despite variations in initial performance and architectural differences among the three models, the generalization capability of our approach, focusing on spatial relationships between objects in the environment, is evident.

4.5.13/ Quantitative analysis of RCC interest

In this section, we aim to show the effectiveness of the proposed approach in improving the models ability to learn and understand knowledge about spatial relationships (RCC-8) through an analytical study. To perform such an analysis, we performed additional experiments. Specifically, we focus on the Cityscapes validation set, considering three

key outcomes: the ground truth of panoptic segmentation, the panoptic segmentation prediction maps of the “EfficientPS-b4" model, and the predictions of the “EfficientPS-b4-RCC" model.

Our analysis involved calculating the delta or absolute value of the difference between the percentages of each pair representing the existence of specific RCC relations. For example, let us consider the pair of regions (road, car). In the ground truth, this pair exists with a percentage of 72%, considering the Partially Overlapping (PO) relationship between the regions classified as road and cars. In the prediction of “EfficientPS-b4", the percentage is 65%, and in the predictions of “EfficientPS-b4-RCC", it is 68%.

The calculated delta reveals that “EfficientPS-b4-RCC" has learned the relationship PO more effectively for the connection between road and cars compared to “EfficientPS-b4". Specifically, the delta between the ground truth and “EfficientPS-b4-RCC" is 4%, while the delta between the ground truth and EfficientPS-b4 is 7%. This suggests that our model, “EfficientPS-b4-RCC", shows better performance in capturing this specific relationship between this pair of regions.

We extended this analytical process to all 19 classes of the Cityscapes dataset, considering the two most frequent relationships: PO and DC (Disconnected). The general conclusion from the two analyses regarding PO and DC remains consistent. It highlights the effectiveness of the proposed approach in improving the models understanding of spatial relationships. The results demonstrate the value and usefulness of incorporating knowledge into the model concerning the spatial connections between pairs of regions. Further details of the conducted analysis are provided in the Appendix A.

4.6/ CONCLUSION

In conclusion, we propose a new informed DL approach as part of hybrid AI, to enhance the performance of DNNs for panoptic segmentation. By integrating prior knowledge into the DL networks, specifically focusing on spatial relationships between objects, our approach offers significant improvements. The integration of this additional knowledge allows the models to gain a deeper understanding of the scene beyond the visual cues present in the images. This integration enhances the models performance and accuracy by enabling them to capture complex object relationships, resolve ambiguities, and overcome panoptic segmentation challenges.

Our approach offers several contributions, including the introduction of a new training methodology, the development of a new loss function, and the validation and evaluation of the proposed approach on various urban scene datasets. The results of our experiments and evaluations consistently show that the proposed approach outperforms the

SOTA and achieves better results with respect to Panoptic Quality metrics (PQ). Moreover, we demonstrate the ability of the model to generalize based on the results of the proposed approach in unseen datasets. The model trained with the additional knowledge shows improved performance even in challenging datasets. This suggests that the model has learned to understand the general context of urban environments and to apply its knowledge effectively across different datasets.

By incorporating meaningful knowledge during the training process, the proposed approach enables the model to better understand the context of the target environment. This leads to better performance and accurate decision-making. The significance of integrating additional knowledge is not limited to panoptic segmentation alone, it extends to other computer vision tasks where understanding context is important. As part of our future work, we aim to enhance the panoptic segmentation results by introducing a local loss function that specifically targets problematic regions. The goal is to provide the network with more precise and explicit knowledge transfer. Additionally, we aim to integrate other types of knowledge, beyond RCC-8, to further enhance the panoptic segmentation.

5

CONCLUSION

5.1/ THESIS SUMMARY

In the computer vision domain, the integration of Deep Neural Networks (DNNs) with Knowledge-Based Systems (KBS) has primarily focused on post-processing and validation, leaving room for improvement in effectively incorporating knowledge into the DNN training process. The challenge lies in extending beyond simple verification and finding innovative ways to integrate prior knowledge into DNN training, using context-aware knowledge and adaptable computer vision DNNs.

Moreover, while existing approaches have demonstrated effectiveness in various computer vision tasks, their application in autonomous driving has some limitations due to the dynamic nature of urban environments. Challenges such as changing weather conditions, moving objects, and unpredictable pedestrian behavior to cite just a few make it intricate to pre-define precise meaningful knowledge for integration into DNNs.

In behave to these challenges, this thesis proposed novel approaches to enhance DNN performance by incorporating knowledge for autonomous driving context. The focus goes beyond achieving superior computer vision task results, expanding on speeding up the training process, reducing resource requirements, optimizing data utilization, and especially building strong DNNs that can understand the general context of the urban environment. Three fundamental questions guided the research: defining meaningful knowledge for a specific task, properly representing this knowledge, and defining the optimal integration strategy into the DNN process.

The research addressed these questions by focusing on two computer vision tasks: monocular depth estimation and panoptic segmentation in urban environments. Monocular depth estimation plays a crucial role in understanding the three-dimensional aspects of the urban environment, which is essential for safe navigation, while panoptic segmentation gives a holistic and complete view of the environment which aids in decision-making.

For monocular depth estimation, we proposed an approach based on knowledge ex-

tracted from ontology reasoning to enhance the performance of DNNs for MDE. The proposed system extracts monocular cues based on human-like reasoning performed on an ontology representing various knowledge of the urban environment. The extracted monocular cues are fed to the model as additional inputs to improve the training process. Experimental validation on diverse datasets of urban environments, using both basic and powerful models, demonstrates that models trained on monocular cues maps consistently outperform other state-of-the-art models. In the second work, an informed deep learning approach was introduced to augment the performance of DNNs for panoptic segmentation. This approach integrates prior knowledge, with a specific focus on spatial relationships between objects. The innovation in this work lies in representing the knowledge as a new loss function to ensure the integration of spatial relationship information directly in the training process.

In conclusion, our research introduces various strategies for the representation and integration of knowledge into DNNs. The first work employs ontology representation and direct integration as input, focusing on MDE. This choice aligns with the pre-acquired knowledge used in human perception to estimate depth using only one eye. By imitating the human depth estimation process, we enhanced DNN performance, demonstrating the importance of knowledge integration. The second contribution focuses on panoptic segmentation, using spatial relationships represented as a specialized loss function integrated during training iterations. The choice of spatial relationships addresses the challenges posed by complex spatial layouts in urban scenarios, where standard models often struggle. Through this strategic representation and integration, we successfully improved the network understanding of intricate spatial relationships, leading to enhanced panoptic segmentation accuracy. Ultimately, our findings underscore the importance of selecting meaningful knowledge aligned with the target task. The choice of monocular cues for MDE and spatial relationships for panoptic segmentation proved the significance of thoughtful knowledge transfer in building more context-aware DNNs.

5.2/ FUTURE WORK AND PERSPECTIVES

Looking toward future work, one key perspective involves introducing a joint learning strategy through a comprehensive global approach. This methodology aims to enhance both monocular depth estimation and panoptic segmentation within a single framework. To achieve this, we propose merging the two distinct methodologies presented in this thesis. First, incorporating knowledge extracted from an ontology as input to a DNN to enhance monocular depth estimation. Second, enhancing panoptic segmentation by integrating knowledge represented as a loss function. The goal is to capitalize on the improved model performance achieved in monocular depth estimation to reinforce and assist the panop-

tic segmentation, and vice versa. This approach is designed to leverage the strengths of each of the proposed knowledge integration methodologies to mutually enhance the performance of the DNNs. This promising direction holds the potential to further advance the combination of KBS and DNNs for more accurate and improved results.

The second perspective involves integrating knowledge at another level of the process. As we presented in the state-of-the-art section (Section 2), we categorized the integration levels of the KBS into DNNs in three stages: as input to the model, during the training of the DNNs, or at the last stages. During this thesis, we investigated two ways of combining KBS with DNNs, namely representing the knowledge as images and feeding them as input to the model, and representing the knowledge as a loss function. The latter can be considered as an integration during the training process, as the knowledge integration is a part of the optimization process of the DNN. One key perspective in this context is to integrate knowledge into the basic architecture of the DNN. In this case, we aim to represent the knowledge and integrate it into the framework in a suitable way to interact with and analyze the output features at different stages of the DNN layers. We believe that this integration stage would provide more information to the DNN, thereby enhancing their performance, speeding up their training, and enabling better characteristics learning that leverage knowledge to perform the target task.

The third proposal consists in exploring other approaches to combine KBS and DNNs. In this thesis, the primary focus was on integrating KBS into DNNs, where the knowledge serves the deep neural networks. Another interesting perspective is to delve deeper into this idea and shift towards joint learning. However, unlike the previous perspective on simultaneously enhancing the performance of multiple computer vision tasks, this perspective aims to elevate the capabilities and performance of both DNN and KBS collectively. The proposed bidirectional learning strategy seeks to further develop this concept, allowing for a dynamic interaction between the two modules or components. The objective is not only to enhance the performance of both DNN and KBS but also to create a strong training relationship where each benefits from the strengths of the other. We believe that adopting a bidirectional learning approach would lead to improved, more precise, and updated knowledge that aligns more closely with the specific needs of the target task. Simultaneously, we anticipate that the DNN would produce enhanced results, given its collaboration with a more rich and adaptable KBS.

5.3/ PUBLICATIONS

Peer-reviewed journals :

- Benkirane, F. E., Crombez, N., Ruichek, Y., & Hilaire, V. (2023). **Integration of on-**

tology reasoning-based monocular cues in deep learning modeling for single image depth estimation in urban driving scenarios. Knowledge-Based Systems, 260, 110184. Impact factor: 8.8

- Benkirane, F. E., Crombez, N., Hilaire, V., & Ruichek, Y. (2023). **Hybrid AI for panoptic segmentation: An informed deep learning approach with integration of prior spatial relationships knowledge** Computer Vision and Image Understanding. Impact factor: 4.5

Peer-reviewed international conferences:

- Benkirane, F. E., Crombez, N., Hilaire, V., & Ruichek, Y. (2022, October). **Depth Estimation Using Deep Learning Guided By Ontology Reasoning-Based Monocular Cues.** In IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society (pp. 1-6). IEEE.

Peer-reviewed national conferences:

- Benkirane, F. E., Crombez, N. C. N., Hilaire, V., & Ruichek, Y. (2023, May). **Intégration de connaissances sur les relations spatiales dans la fonction de perte d'un réseau de neurones pour la segmentation panoptique.** In ORASIS 2023.

BIBLIOGRAPHY

[1] A. Faisal, T. Yigitcanlar, M. Kamruzzaman, A. Paz, Mapping two decades of autonomous vehicle research: A systematic scientometric analysis, Journal of Urban Technology 28 (3-4) (2021) 45–74.

[2] S. Naumov, D. R. Keith, C. H. Fine, Unintended consequences of automated vehicles and pooling for urban transportation systems, Production and Operations Management 29 (5) (2020) 1354–1371.

[3] D. Milakis, B. van Wee, Implications of vehicle automation for accessibility and social inclusion of people on low income, people with physical and sensory disabilities, and older people, in: Demand for emerging transportation systems, Elsevier, 2020, pp. 61–73.

[4] A. Faisal, M. Kamruzzaman, T. Yigitcanlar, G. Currie, Understanding autonomous vehicles, Journal of transport and land use 12 (1) (2019) 45–72.

[5] P. H. Feiler, B. Lewis, S. Vestal, E. Colbert, An overview of the sae architecture analysis & design language (aadl) standard: A basis for model-based architecture-driven embedded systems engineering, in: IFIP World Computer Congress, TC 2, Springer, 2004, pp. 3–15.

[6] J. Fayyad, M. A. Jaradat, D. Gruyer, H. Najjaran, Deep learning sensor fusion for autonomous vehicle perception and localization: A review, Sensors 20 (15) (2020) 4220.

[7] C. Katrakazas, M. Quddus, W.-H. Chen, L. Deka, Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions, Transportation Research Part C: Emerging Technologies 60 (2015) 416–442.

[8] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, M. H. Ang Jr, Perception, planning, control, and coordination for autonomous vehicles, Machines 5 (1) (2017) 6.

[9] F. Liu, Z. Lu, X. Lin, Vision-based environmental perception for autonomous driving, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering (2022) 09544070231203059.

[10] G. Stockman, L. G. Shapiro, Computer vision, Prentice Hall PTR, 2001.

[11] E. Karami, M. Shehata, A. Smith, Image identification using sift algorithm: performance analysis against different image deformations, arXiv preprint arXiv:1710.02728.

[12] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: 9th European Conference on Computer Vision, Vol. Part I 9, Springer, 2006, pp. 404–417.

[13] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: 9th European Conference on Computer Vision, Vol. Part I 9, Springer, 2006, pp. 430–443.

[14] H. M. Wallach, Topic modeling: beyond bag-of-words, in: Proceedings of the 23rd international conference on Machine learning, 2006, pp. 977–984.

[15] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: analysis, applications, and prospects, IEEE transactions on neural networks and learning systems.

[16] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, J. Walsh, Deep learning vs. traditional computer vision, in: Proceedings of the Computer Vision Conference (CVC), Vol. 1 1, Springer, 2020, pp. 128–144.

[17] X. Shi, Z. Zheng, Y. Zhou, H. Jin, L. He, B. Liu, Q.-S. Hua, Graph processing on gpus: A survey, ACM Computing Surveys (CSUR) 50 (6) (2018) 1–35.

[18] C. Room, Tensor processing unit, machine learning 15 (54) (2021) 13.

[19] S. Vicente, J. Carreira, L. Agapito, J. Batista, Reconstructing pascal voc, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 41–48.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[21] B. César, Human perception inside of a self-driving robotic car, IPSI Transactions on Advanced Research 17 (2) (2021) 50–56.

[22] D. Omeiza, K. Kollnig, H. Web, M. Jirotka, L. Kunze, Why not explain? effects of explanations on human perceptions of autonomous driving, in: 2021 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO), IEEE, 2021, pp. 194–199.

[23] J. Wang, Y. Li, R. X. Gao, F. Zhang, Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability, Journal of Manufacturing Systems 63 (2022) 381–391.

[24] K. S. M. H. Ibrahim, Y. F. Huang, A. N. Ahmed, C. H. Koo, A. El-Shafie, A review of the hybrid artificial intelligence and optimization modelling of hydrological streamflow forecasting, Alexandria Engineering Journal 61 (1) (2022) 279–303.

[25] T. Rajaee, S. Khani, M. Ravansalar, Artificial intelligence-based single and hybrid models for prediction of water quality in rivers: A review, Chemometrics and Intelligent Laboratory Systems 200 (2020) 103978.

[26] M. Alirezaie, M. Längkvist, M. Sioutis, A. Loutfi, Semantic referee: A neural-symbolic framework for enhancing geospatial semantic segmentation, Semantic Web 10 (5) (2019) 863–880.

[27] C. Mazo, E. Alegre, M. Trujillo, Using an ontology of the human cardiovascular system to improve the classification of histological images, Scientific Reports 10 (1) (2020) 12276.

[28] M. Alirezaie, M. Längkvist, M. Sioutis, A. Loutfi, A symbolic approach for explaining errors in image classification tasks, in: IJCAI Workshop on Learning and Reasoning. Stockholm, Sweden, 2018.

[29] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[32] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[34] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, Computational Social Networks 6 (1) (2019) 1–23.

[35] N. F. Noy, D. L. McGuinness, et al., Ontology development 101: A guide to creating your first ontology (2001).

[36] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical owl-dl reasoner, Journal of Web Semantics 5 (2) (2007) 51–53.

[37] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, Hermit: an owl 2 reasoner, Journal of automated reasoning 53 (2014) 245–269.

[38] V. Haarslev, K. Hidde, R. Möller, M. Wessel, The racerpro knowledge representation and reasoning system, Semantic Web 3 (3) (2012) 267–277.

[39] D. Tsarkov, I. Horrocks, Fact++ description logic reasoner: System description, in: International joint conference on automated reasoning, Springer, 2006, pp. 292–297.

[40] A. Ameen, K. U. R. Khan, B. P. Rani, Reasoning in semantic web using jena, Computer Engineering and Intelligent Systems 5 (4) (2014) 39–47.

[41] R. Swick, Resource description framework (rdf) model and syntax specification, URL http://www. w3. org/TR/1999/REC-rdfsyntax-19990222.

[42] J. Broekstra, A. Kampman, F. Van Harmelen, Sesame: A generic architecture for storing and querying rdf and rdf schema, in: International semantic web conference, Springer, 2002, pp. 54–68.

[43] D. L. McGuinness, F. Van Harmelen, et al., Owl web ontology language overview, W3C recommendation 10 (10) (2004) 2004.

[44] D. Beckett, B. McBride, Rdf/xml syntax specification (revised), W3C recommendation 10 (2.3).

[45] M. Grabmüller, P. Hofstedt, Turtle: A constraint imperative programming language, in: International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 2003, pp. 185–198.

[46] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, N. Lindström, Json-ld 1.1, W3C Recommendation, Jul.

[47] A. Seaborne, E. Prud'hommeaux, Sparql query language for rdf, W3C recommendation.

[48] Y. Li, S. Ouyang, Y. Zhang, Combining deep learning and ontology reasoning for remote sensing image semantic segmentation, Knowledge-Based Systems 243 (2022) 108469.

[49] S. Andrés, D. Arvor, I. Mougenot, T. Libourel, L. Durieux, Ontology-based classification of remote sensing images using spectral rules, Computers & Geosciences 102 (2017) 158–166.

[50] D. Triboan, L. Chen, F. Chen, Z. Wang, Semantic segmentation of real-time sensor data stream for complex activity recognition, Personal and Ubiquitous Computing 21 (2017) 411–425.

[51] Y. Breux, S. Druon, R. Zapata, From perception to semantics: An environment representation model based on human-robot interactions, in: 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), IEEE, 2018, pp. 672–677.

[52] J. Liu, X. Zhang, Y. Li, J. Wang, H.-J. Kim, Deep learning-based reasoning with multi-ontology for iot applications, IEEE Access 7 (2019) 124688–124701.

[53] P. Hohenecker, T. Lukasiewicz, Deep learning for ontology reasoning, CoRR.

[54] F. N. Al-Aswadi, H. Y. Chan, K. H. Gan, Automatic ontology construction from text: a review from shallow to deep learning trend, Artificial Intelligence Review 53 (2020) 3901–3928.

[55] N. Phan, D. Dou, H. Wang, D. Kil, B. Piniewski, Ontology-based deep learning for human behavior prediction with explanations in health social networks, Information sciences 384 (2017) 298–313.

[56] J. E. Ball, D. T. Anderson, C. S. Chan, Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community, Journal of applied remote sensing 11 (4) (2017) 042609–042609.

[57] G. Cheng, J. Han, X. Lu, Remote sensing image scene classification: Benchmark and state of the art, Proceedings of the IEEE 105 (10) (2017) 1865–1883.

[58] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: 21st International Conference on Artificial Neural Networks, Vol. Part I 21, Springer, 2011, pp. 52–59.

[59] M. Alirezaie, A. Kiselev, M. Längkvist, F. Klügl, A. Loutfi, An ontology-based reasoning framework for querying satellite images for disaster monitoring, Sensors 17 (11) (2017) 2545.

[60] C. Mazo, L. Salazar, O. Corcho, M. Trujillo, E. Alegre, A histological ontology of the human cardiovascular system, Journal of biomedical semantics 8 (2017) 1–15.

[61] C. Goutte, E. Gaussier, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation, in: European conference on information retrieval, Springer, 2005, pp. 345–359.

[62] S. Albawi, T. A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network, in: International Conference on Engineering and Technology (ICET), Ieee, 2017, pp. 1–6.

[63] M. K. Sarker, N. Xie, D. Doran, M. Raymer, P. Hitzler, Explaining trained neural networks with semantic web technologies: First steps, arXiv preprint arXiv:1710.04324.

[64] Sumo ontology, http://www.adampease.org/OP/.

[65] R. T. Icarte, J. A. Baier, C. Ruz, A. Soto, How a general-purpose commonsense ontology can improve performance of learning-based image retrieval, arXiv preprint arXiv:1705.08844.

[66] S. Mukherjee, S. Joshi, Sentiment aggregation using conceptnet ontology, in: Proceedings of the sixth international joint conference on natural language processing, 2013, pp. 570–578.

[67] X. Ding, Y. Luo, Q. Li, Y. Cheng, G. Cai, R. Munnoch, D. Xue, Q. Yu, X. Zheng, B. Wang, Prior knowledge-based deep learning method for indoor object recognition and application, Systems Science & Control Engineering 6 (1) (2018) 249–257.

[68] S. Ilyas, H. U. Rehman, A deep learning based approach for precise video tagging, in: 15th International Conference on Emerging Technologies (ICET), IEEE, 2019, pp. 1–6.

[69] P. Rodríguez, M. A. Bautista, J. Gonzalez, S. Escalera, Beyond one-hot encoding: Lower dimensional target embedding, Image and Vision Computing 75 (2018) 21–31.

[70] Y. Li, S. Ouyang, Y. Zhang, Collaboratively boosting data-driven deep learning and knowledge-guided ontological reasoning for semantic segmentation of remote sensing imagery, arXiv preprint arXiv:2010.02451.

[71] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: 18th International Conference of Medical Image Computing and Computer-Assisted Intervention, Vol. Part III 18, Springer, 2015, pp. 234–241.

[72] K.-H. Lee, G. Ros, J. Li, A. Gaidon, Spigan: Privileged adversarial learning from simulation, in: International Conference on Learning Representations, 2018.

[73] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, J. Garcke, Combining machine learning and simulation to a hybrid modelling approach: Current and future directions, in: 18th International Symposium on Intelligent Data Analysis, Springer, 2020, pp. 548–560.

[74] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR, 2015, pp. 1180–1189.

[75] A. Khadka, P. Remagnino, V. Argyriou, Synthetic crowd and pedestrian generator for deep learning problems, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 4052–4056.

[76] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, S. Savarese, Social lstm: Human trajectory prediction in crowded spaces, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 961–971.

[77] D. Helbing, P. Molnar, Social force model for pedestrian dynamics, Physical review E 51 (5) (1995) 4282.

[78] B. Romera-Paredes, P. Torr, An embarrassingly simple approach to zero-shot learning, in: International conference on machine learning, PMLR, 2015, pp. 2152–2161.

[79] J. Chen, F. Lécué, Y. Geng, J. Z. Pan, H. Chen, Ontology-guided semantic composition for zero-shot learning, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, Vol. 17, 2020, pp. 850–854.

[80] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 31, 2017.

[81] S. Palazzo, F. Murabito, C. Pino, F. Rundo, D. Giordano, M. Shah, C. Spampinato, Exploiting structured high-level knowledge for domain-specific visual classification, Pattern Recognition 112 (2021) 107806.

[82] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[83] D. Heckerman, A tutorial on learning with bayesian networks, Innovations in Bayesian networks: Theory and applications (2008) 33–82.

[84] Z. Hu, X. Ma, Z. Liu, E. Hovy, E. Xing, Harnessing deep neural networks with logic rules, arXiv preprint arXiv:1603.06318.

[85] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531.

[86] K. Ganchev, J. Graça, J. Gillenwater, B. Taskar, Posterior regularization for structured latent variable models, The Journal of Machine Learning Research 11 (2010) 2001–2049.

[87] A. Antonucci, G. P. R. Papini, L. Palopoli, D. Fontanelli, Generating reliable and efficient predictions of human motion: A promising encounter between physics and neural networks, arXiv preprint arXiv:2006.08429.

[88] Z. Zhang, L. Jia, Direction-decision learning based pedestrian flow behavior investigation, IEEE Access 8 (2020) 15027–15038.

[89] P. Kothari, B. Sifringer, A. Alahi, Interpretable social anchors for human trajectory forecasting in crowds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15556–15566.

[90] Y. Li, R. Zemel, M. Brockschmidt, D. Tarlow, Gated graph sequence neural networks, in: Proceedings of ICLR, 2016.

[91] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE transactions on neural networks 20 (1) (2008) 61–80.

[92] C.-Y. Chuang, J. Li, A. Torralba, S. Fidler, Learning to act properly: Predicting and explaining affordances from images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 975–983.

[93] C. Ye, Y. Yang, R. Mao, C. Fermüller, Y. Aloimonos, What can i do around here? deep functional scene understanding for cognitive robots, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 4604–4611.

[94] J. Sawatzky, Y. Souri, C. Grund, J. Gall, What object should i use?-task driven object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7605–7614.

[95] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, S. Fidler, Situation recognition with graph neural networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 4173–4182.

[96] X. Chen, L.-J. Li, L. Fei-Fei, A. Gupta, Iterative visual reasoning beyond convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7239–7248.

[97] E. Muller-Budack, M. Springstein, S. Hakimov, K. Mrutzek, R. Ewerth, Ontology-driven event type classification in images, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2021, pp. 2928–2938.

[98] P. Sitikhu, K. Pahi, P. Thapa, S. Shakya, A comparison of semantic similarity methods for maximum human interpretability, in: Artificial Intelligence for Transforming Business and society (AITB), Vol. 1, IEEE, 2019, pp. 1–4.

[99] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[100] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (11) (1995) 39–41.

[101] A. Colmerauer, An introduction to prolog iii, Communications of the ACM 33 (7) (1990) 69–90.

[102] M. Zand, S. Doraisamy, A. A. Halin, M. R. Mustaffa, Ontology-based semantic image segmentation using mixture models and multiple crfs, IEEE Transactions on Image Processing 25 (7) (2016) 3233–3248.

[103] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, S. Rudolph, et al., Owl 2 web ontology language primer, W3C recommendation 27 (1) (2009) 123.

[104] R. M. Neal, Markov chain sampling methods for dirichlet process mixture models, Journal of computational and graphical statistics 9 (2) (2000) 249–265.

[105] C. Sutton, A. McCallum, et al., An introduction to conditional random fields, Foundations and Trends® in Machine Learning 4 (4) (2012) 267–373.

[106] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, Integrating scientific knowledge with machine learning for engineering and environmental systems, ACM Computing Surveys 55 (4) (2022) 1–37.

[107] M. Silvestri, M. Lombardi, M. Milano, Injecting domain knowledge in neural networks: a controlled experiment on a constrained problem, in: Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 18th International Conference, Springer, 2021, pp. 266–282.

[108] M. Diligenti, M. Gori, C. Sacca, Semantic-based regularization for learning and inference, Artificial Intelligence 244 (2017) 143–165.

[109] F. Rossi, P. Van Beek, T. Walsh, Handbook of constraint programming, Elsevier, 2006.

[110] Z. Yang, J.-L. Wu, H. Xiao, Enforcing deterministic constraints on generative adversarial networks for emulating physical systems, arXiv preprint arXiv:1911.06671.

[111] J.-L. Wu, K. Kashinath, A. Albert, D. Chirila, H. Xiao, et al., Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems, Journal of Computational Physics 406 (2020) 109209.

[112] R. Cang, H. Li, H. Yao, Y. Jiao, Y. Ren, Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model, Computational Materials Science 150 (2018) 212–221.

[113] A. Daw, A. Karpatne, W. D. Watkins, J. S. Read, V. Kumar, Physics-guided neural networks (pgnn): An application in lake temperature modeling, in: Knowledge Guided Machine Learning, Chapman and Hall/CRC, 2022, pp. 353–372.

[114] X. Jia, B. Lin, J. Zwart, J. Sadler, A. Appling, S. Oliver, J. Read, Graph-based reinforcement learning for active learning in real time: An application in modeling river networks, in: Proceedings of the SIAM International Conference on Data Mining (SDM), SIAM, 2021, pp. 621–629.

[115] J. S. Read, X. Jia, J. Willard, A. P. Appling, J. A. Zwart, S. K. Oliver, A. Karpatne, G. J. Hansen, P. C. Hanson, W. Watkins, et al., Process-guided deep learning predictions of lake water temperature, Water Resources Research 55 (11) (2019) 9173–9190.

[116] M. L. Thompson, M. A. Kramer, Modeling chemical processes using prior knowledge and neural networks, AIChE Journal 40 (8) (1994) 1328–1340.

[117] J. Bornia, S. A. Mahmoudi, A. Frihida, P. Manneback, Towards a semantic video analysis using deep learning and ontology, in: 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech), IEEE, 2018, pp. 1–6.

[118] M. et al., Deep learning for monocular depth estimation: A review., Neurocomputing.

[119] M. et al., Real-time pose and shape reconstruction of two interacting hands with a single depth camera, Transactions on Graphics (TOG) 38 (4) (2019) 1–13.

[120] L. Zou, Y. Li, A method of stereo vision matching based on opencv, in: International Conference on Audio, Language and Image Processing, IEEE, 2010, pp. 185–190.

[121] Y.-M. Tsai, Y.-L. Chang, L.-G. Chen, Block-based vanishing line and vanishing point detection for 3d scene reconstruction, in: International symposium on intelligent signal processing and communications, IEEE, 2006, pp. 586–589.

[122] Y.-K. Wang, C.-T. Fan, C.-W. Chang, Accurate depth estimation for image defogging using markov random field, in: International Conference on Graphic and Image Processing, Vol. 8768, International Society for Optics and Photonics, 2013, p. 87681Q.

[123] A. Bosch, A. Zisserman, X. Munoz, Image classification using random forests and ferns, in: 11th international conference on computer vision, IEEE, 2007, pp. 1–8.

[124] Z. et al., An algorithm of single image depth estimation based on mrf model, in: International Conference on Wireless and Satellite Systems, Springer, 2019, pp. 198–207.

[125] K. et al., Improved depth map estimation from stereo images based on hybrid method., Radioengineering 21 (1).

[126] R. et al., Disparity estimation from stereo images, Procedia engineering 38 (2012) 462–472.

[127] Z. et al., Stereo matching by training a convolutional neural network to compare image patches., J. Mach. Learn. Res. 17 (1) (2016) 2287–2318.

[128] L. et al., Binocular light-field: Imaging theory and occlusion-robust depth perception application, Transactions on Image Processing 29 (2019) 1628–1640.

[129] F. et al., Cam-convs: camera-aware multi-scale convolutions for single-view depth, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11826–11835.

[130] G. et al., Unsupervised cnn for single view depth estimation: Geometry to the rescue, in: European conference on computer vision, Springer, 2016, pp. 740–756.

[131] P. Chakravarty, P. Narayanan, T. Roussel, Gen-slam: Generative modeling for monocular simultaneous localization and mapping, in: International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 147–153.

[132] T. v. Dijk, G. d. Croon, How do neural networks see depth in single images?, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2183–2191.

[133] J. Hu, Y. Zhang, T. Okatani, Visualization of convolutional neural networks for monocular depth estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3869–3878.

[134] G. et al., Digging into self-supervised monocular depth estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3828–3838.

[135] C. Godard, O. Mac Aodha, G. J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 270–279.

[136] B. et al., Adabins: Depth estimation using adaptive bins, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4009–4018.

[137] B. et al., Exploiting semantic information and deep matching for optical flow, in: European Conference on Computer Vision, Springer, 2016, pp. 154–170.

[138] K. et al., Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance, in: European Conference on Computer Vision, Springer, 2020, pp. 582–600.

[139] M. Ochs, A. Kretz, R. Mester, Sdnet: Semantically guided depth estimation network, in: German conference on pattern recognition, Springer, 2019, pp. 288–302.

[140] C. Hong, J. Yu, J. Zhang, X. Jin, K.-H. Lee, Multimodal face-pose estimation with multitask manifold deep learning, IEEE transactions on industrial informatics 15 (7) (2018) 3952–3961.

[141] C. Hong, J. Yu, J. You, X. Chen, D. Tao, Multi-view ensemble manifold regularization for 3d object recognition, Information sciences 320 (2015) 395–405.

[142] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Transactions on Industrial Electronics 62 (6) (2014) 3742–3751.

[143] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE transactions on image processing 24 (12) (2015) 5659–5670.

[144] H. et al., Swrl: A semantic web rule language combining owl and ruleml, W3C Member submission 21 (79) (2004) 1–31.

[145] L. et al., Deeper depth prediction with fully convolutional residual networks, in: Fourth international conference on 3D vision (3DV), IEEE, 2016, pp. 239–248.

[146] H. Fu, M. Gong, C. Wang, K. Batmanghelich, D. Tao, Deep ordinal regression network for monocular depth estimation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2002–2011.

[147] Z. et al., Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[148] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.

[149] T. et al., Supporting collaborative ontology development in protégé, in: International Semantic Web Conference, Springer, 2008, pp. 17–32.

[150] Owlready2 package, https://owlready2.readthedocs.io/en/v0.32/.

[151] Adabins code, https://github.com/shariqfarooq123/AdaBins.

[152] D. Soydaner, A comparison of optimization algorithms for deep learning, International Journal of Pattern Recognition and Artificial Intelligence 34 (13) (2020) 2052013.

[153] L. et al., Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[154] C. et al., The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213–3223.

[155] V. Guizilini, R. Hou, J. Li, R. Ambrus, A. Gaidon, Semantically-guided representation learning for self-supervised monocular depth, in: International Conference on Learning Representations, 2019.

[156] G. et al., Vision meets robotics: The ktti dataset, The International Journal of Robotics Research 32 (11) (2013) 1231–1237.

[157] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, Advances in neural information processing systems 27.

[158] A. Saxena, S. Chung, A. Ng, Learning depth from single monocular images, Advances in neural information processing systems 18.

[159] W. et al., Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 541–550.

[160] F. Saeedan, S. Roth, Boosting monocular depth with panoptic segmentation maps, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 3853–3862.

[161] H. et al., The apolloscape dataset for autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 954–960.

[162] D. Song, W.-Z. Nie, W.-H. Li, M. Kankanhalli, A.-A. Liu, Monocular image-based 3-d model retrieval: A benchmark, IEEE Transactions on Cybernetics 52 (8) (2021) 8114–8127.

[163] L. et al., Learning depth from single monocular images using deep convolutional neural fields, Transactions on pattern analysis and machine intelligence 38 (10) (2015) 2024–2039.

[164] Y. Kuznietsov, J. Stuckler, B. Leibe, Semi-supervised deep learning for monocular depth map prediction, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6647–6655.

[165] A. Gurram, O. Urfalioglu, I. Halfaoui, F. Bouzaraa, A. M. López, Monocular depth estimation by learning from heterogeneous datasets, in: IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 2176–2181.

[166] G. et al., Monocular depth estimation with affinity, vertical pooling, and label enhancement, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 224–239.

[167] Y. et al., Enforcing geometric constraints of virtual normal for depth prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5684–5693.

[168] S. Gur, L. Wolf, Single image depth estimation trained via depth from defocus cues, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7683–7692.

[169] B. et al., Unsupervised scale-consistent depth and ego-motion learning from monocular video, Advances in neural information processing systems 32.

[170] G. et al., Gluoncv and gluonnlp: deep learning in computer vision and natural language processing., J. Mach. Learn. Res. 21 (23) (2020) 1–7.

[171] Z. et al., Icnet for real-time semantic segmentation on high-resolution images, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 405–420.

[172] X. et al., Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 675–684.

[173] Z. et al., Joint task-recursive learning for semantic segmentation and depth estimation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 235–251.

[174] S. et al., Mlda-net: multi-level dual attention-based network for self-supervised monocular depth estimation, Transactions on Image Processing 30 (2021) 4691–4705.

[175] F. E. Benkirane, N. Crombez, Y. Ruichek, V. Hilaire, Integration of ontology reasoning-based monocular cues in deep learning modeling for single image depth estimation in urban driving scenarios, Knowledge-Based Systems 260 (2023) 110184.

[176] F. E. Benkirane, N. Crombez, V. Hilaire, Y. Ruichek, Depth estimation using deep learning guided by ontology reasoning-based monocular cues, in: IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2022, pp. 1–6.

[177] H. Zhu, C. Yao, Z. Zhu, Z. Liu, Z. Jia, Fusing panoptic segmentation and geometry information for robust visual slam in dynamic environments, in: IEEE 18th International Conference on Automation Science and Engineering (CASE), IEEE, 2022, pp. 1648–1653.

[178] A. Milioto, J. Behley, C. McCool, C. Stachniss, Lidar panoptic segmentation for autonomous driving, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 8505–8512.

[179] O. Zendel, M. Schörghuber, B. Rainer, M. Murschitz, C. Beleznai, Unifying panoptic segmentation for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 21351–21360.

[180] K. et al., Panoptic feature pyramid networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6399–6408.

[181] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, W. Jiang, An end-to-end network for panoptic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 6172–6181.

[182] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 891–898.

[183] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, M. Hebert, An empirical study of context in object detection, in: IEEE Conference on computer vision and Pattern Recognition, 2009, pp. 1271–1278.

[184] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on pattern analysis and machine intelligence 32 (9) (2009) 1627–1645.

[185] J. R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders, Selective search for object recognition, International journal of computer vision 104 (2013) 154–171.

[186] C. Galleguillos, S. Belongie, Context based object categorization: A critical survey, Computer vision and image understanding 114 (6) (2010) 712–722.

[187] H. Hu, J. Gu, Z. Zhang, J. Dai, Y. Wei, Relation networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3588–3597.

[188] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, in: Proceedings of the International Conference on Learning Representations (ICLR), ICLR, 2014, p. 30.

[189] P. Pinheiro, R. Collobert, Recurrent convolutional neural networks for scene labeling, in: International conference on machine learning, PMLR, 2014, pp. 82–90.

[190] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern recognition 77 (2018) 354–377.

[191] G.-B. Huang, Z. Bai, L. L. C. Kasun, C. M. Vong, Local receptive fields based extreme learning machine, IEEE Computational intelligence magazine 10 (2) (2015) 18–29.

[192] L. R. Medsker, Hybrid intelligent systems, Springer Science & Business Media, 2012.

[193] M. Seera, C. P. Lim, A hybrid intelligent system for medical data classification, Expert systems with applications 41 (5) (2014) 2239–2249.

[194] O. Castillo, P. Melin, W. Pedrycz, Hybrid intelligent systems: Analysis and design, Vol. 208, Springer, 2007.

[195] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (6) (2021) 422–440.

[196] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, et al., Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems, IEEE Transactions on Knowledge and Data Engineering 35 (1) (2021) 614–633.

[197] R. Mohan, A. Valada, Efficientps: Efficient panoptic segmentation, International Journal of Computer Vision 129 (5) (2021) 1551–1579.

[198] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, L.-C. Chen, Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 12475–12485.

[199] C.-Y. Chang, S.-E. Chang, P.-Y. Hsiao, L.-C. Fu, Epsnet: efficient panoptic segmentation network with cross-layer attention fusion, in: Proceedings of the Asian Conference on Computer Vision, 2020, p. 33.

[200] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, L.-C. Chen, Axial-deeplab: Stand-alone axial-attention for panoptic segmentation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, Proceedings, Part IV, Springer, 2020, pp. 108–126.

[201] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: 13th European Conference in Computer Vision, Springer, 2014, pp. 740–755.

[202] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: IEEE conference on computer vision and pattern recognition, 2012, pp. 3354–3361.

[203] J. Lazarow, K. Lee, K. Shi, Z. Tu, Learning instance occlusion for panoptic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10720–10729.

[204] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, Proceedings of the AAAI conference on artificial intelligence (2017) 30.

[205] S. R. Bulo, L. Porzi, P. Kontschieder, In-place activated batchnorm for memory-optimized training of dnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5639–5647.

[206] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[207] T.-J. Yang, M. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, L.-C. Chen, Deeperlab: Single-shot image parser, Artificial Intelligence, Communication, Imaging, Navigation, Sensing Systems (2019) 10.

[208] N. Gao, F. He, J. Jia, Y. Shan, H. Zhang, X. Zhao, K. Huang, Panopticdepth: A unified framework for depth-aware panoptic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1632–1642.

[209] Y. Wu, G. Zhang, H. Xu, X. Liang, L. Lin, Auto-panoptic: Cooperative multi-component architecture search for panoptic segmentation, Advances in Neural Information Processing Systems 33 (2020) 20508–20519.

[210] D. De Geus, P. Meletis, G. Dubbelman, Single network panoptic segmentation for street scene understanding, in: IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 709–715.

[211] Y. Chen, G. Lin, S. Li, O. Bourahla, Y. Wu, F. Wang, J. Feng, M. Xu, X. Li, Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 3793–3802.

[212] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.

[213] A. G. Cohn, B. Bennett, J. Gooday, N. M. Gotts, Qualitative spatial representation and reasoning with the region connection calculus, geoinformatica 1 (1997) 275–316.

[214] D. A. Randell, A. G. Cohn, Modelling topological and metrical properties in physical processes., KR 89 (1989) 357–368.

[215] Z. Long, S. Li, On distributive subalgebras of qualitative spatial and temporal calculi, in: International Conference on Spatial Information Theory, Springer, 2015, pp. 354–374.

[216] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1492–1500.

[217] P. et al., Seamless scene segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8277–8286.

[218] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in python, PeerJ 2 (2014) e453.

[219] Y. Gatsoulis, M. Alomari, C. Burbridge, C. Dondrup, P. Duckworth, P. Lightbody, M. Hanheide, N. Hawes, D. Hogg, A. Cohn, et al., Qsrlib: a software library for online acquisition of qualitative spatial relations from video (2016) 30.

[220] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, C. Jawahar, Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments, in: IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1743–1751.

[221] Q. Li, A. Arnab, P. H. Torr, Weakly-and semi-supervised panoptic segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 102–118.

[222] A. Kirillov, R. Girshick, K. He, P. Dollár, Panoptic feature pyramid networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 6399–6408.

[223] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, X. Wang, Attention-guided unified network for panoptic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7026–7035.

[224] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, R. Urtasun, Upsnet: A unified panoptic segmentation network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8818–8826.

[225] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, K. Huang, Ssap: Single-shot instance segmentation with affinity pyramid, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 642–651.

[226] K. Sofiiuk, O. Barinova, A. Konushin, Adaptis: Adaptive instance selection network, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7355–7363.

[227] K. Sirohi, S. Marvi, D. Büscher, W. Burgard, Uncertainty-aware panoptic segmentation, IEEE Robotics and Automation Letters.

[228] Q. Li, X. Qi, P. H. Torr, Unifying training and inference for panoptic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13320–13328.

[229] W. Liu, A. Rabinovich, A. C. Berg, Parsenet: Looking wider to see better, arXiv preprint arXiv:1506.04579.

[230] Y. Li, H. Zhao, X. Qi, L. Wang, Z. Li, J. Sun, J. Jia, Fully convolutional networks for panoptic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 214–223.

[231] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

List of Figures

List of Tables

List of Definitions

AV : Autonomous vehicles

AI : Artificial intelligence

DL : Deep Learning

ML : Machine Learning

SAE : Society of Automotive Engineers

ADAS : Advanced Driver Assistance Systems

LKAS : Lane Keeping Assist Systems

AEB : Automatic Emergency Braking

SIFT : Scale Invariant Feature Transform

SURF : Speeded Up Robust Features

FAST : Features from Accelerated Segment Test

SOTA : State Of The Art

NLP : Natural Language Processing

DNN : Deep Neural Network

GPUs : Graphics Processing Units

TPUs : Tensor Processing Units

MDE : Monocular Depth Estimation

KBS : Knowledge-Based Systems

CNN : Convolutional Neural Network

CAE : Convolutional AutoEncoders

RDF : Resource Description Framework

CBF : Collaborative Boosting Framework

RS : Remote Sensing

SFM : Social Force Model

AIC : Animal Image Classification

VQA : Visual Question Answering

OWL : Ontology Web Language

CRFs : Conditional Random Fields

SUMO : Suggested Upper Merged Ontology

FoVs : Frames from Videos

PI : Privileged Information

P : Privileged Network

ZSL : Zero-Shot Learning

GGNNs : Gated Graph Neural Networks

GNNs : Gated Neural Networks

PLS : Partial Latin Square

CP : Constraint Programming

SBR : Semantic Based Regularization

GANs : Generative Adversarial Networks

OBSIS : Ontology-Based Semantic Image Segmentation

MRF : Markov Random Field

PHOG : Pyramid Histogram of Oriented Graph

MDA : Multimodal Deep Autoencoder

BP-NN : Backpropagation-Neural Network

LRR : Low-Rank Representation

ASPP : Atrous Spatial Pyramid Pooling

BANet : Bidirectional Aggregation Network

A

RCC-8 ANALYSIS

Table A.2 shows the analysis of regions regarding the RCC relation PO. In this table, $\Delta 1$ represents the difference between the ground truth and the model “EfficientPS-b4" predictions, and $\Delta 2$ represents the difference between the ground truth and the model “EfficientPS-b4-RCC" predictions. All values are in percentages. We have highlighted in green the results indicating that our model “EfficientPS-b4-RCC" behaves better than the baseline “EfficientPS-b4", specifically when $\Delta 2$ is smaller than $\Delta 1$. The yellow cells indicate results where the baseline and our model behave the same in terms of the number of errors, precisely when $\Delta 2$ equals $\Delta 1$. Finally, the cells highlighted in red correspond to results where $\Delta 2$ is greater than $\Delta 1$, indicating that our proposed model is less better than the baseline. The cells in grey represent pairs of regions that we did not consider due to the reversible nature of the relationship PO. In other words, if region A is PO with region B, it is equivalent to saying that region B is PO with region A. To avoid redundancy, we reported the results for one example of such pairs.

In table A.2, we first observe that most cells are highlighted in green, indicating that the model trained according to the proposed approach successfully learned the spatial relationship PO for the majority of pairs of regions. While there are some red cells where our model performs less better than the baseline, these instances are minor compared to successful ones (green cells). Additionally, when our model outperforms the baseline (green cells), this is often with high percentage values, while the percentages are slight in the opposite case (red cells). Furthermore, there are instances where $\Delta 2$, the difference between our model “EfficientPS-b4-RCC" and the ground truth, reached 0%, indicating an exact matching with the ground truth. Examples of such relationships include (traffic light, sidewalk), (fence, wall), and (traffic sign, motorcycle). This suggests that the model successfully learned situations where these types of regions are connected or not by the relationship PO.

Table A.1 shows the analysis of regions regarding the RCC relation DC. The spatial relationship DC (Disconnected) is recognized as one of the most frequent spatial relationships in the environment. In the images, objects often have a disconnected relationship

with all other objects that are not explicitly connected with them. This characteristic underscores the widespread occurrence of independent entities within the environment, highlighting the significance of the Disconnected spatial relationship in describing the layout and arrangement of objects in images.

Table A.1 presents the delta percentage between the outcomes of the models (ours and baseline) and the ground truth considering the pair of regions for the DC relationship. Once again, most results demonstrate that the model trained with the proposed RCC loss function succeeds in outperforming the baseline by accurately recognizing the DC relationship between objects. This indicates that the model developed an enhanced ability to define whether a pair of objects could be linked with DC relationship, which is mainly important, especially for a spatial relationship that is dominant in different environmental contexts.

In summary, our analysis underscores the effectiveness of the proposed approach in helping the models understanding of spatial relationships. The presented tables, focusing on the two key spatial relationships, clearly illustrate the usefulness of integrating knowledge into the model regarding how pairs of regions are spatially connected. While some pairs show more improvement than others, this variation could be linked to the distribution and complexity of class categories in the dataset as well as the frequency of pairs of regions according to certain spatial relationships. However, the overall results indicate improvement in the majority of classes, directly increasing the model overall task performance.

Δ2 \ Δ1	road	sidewalk	building	pole	traffic sign	traffic light	vegetation	sky	fence	wall	terrain	person	rider	car	bicycle	bus	truck	train	motorcycle
road	5 3	13 8	15 11	10 9	15 13	13 3	7 6	4 2	5 1	9 6	13 10	12 7	16 11	20 23	12 9	14 11	18 15	25 23	12 11
sidewalk		3 0	13 10	12 10	25 18	13 8	6 4	4 2	3 5	8 5	12 11	7 5	7 4	12 10	5 5	5 2	5 1	22 21	4 5
building			6 5	11 6	9 8	11 7	12 10	8 3	6 4	5 3	3 0	13 9	10 8	5 6	12 11	13 12	13 12	10 10	18 15
pole				9 7	22 20	28 22	8 5	5 2	9 7	11 10	12 7	14 11	5 3	22 19	8 6	16 14	7 5	28 25	20 17
traffic sign					19 16	9 11	11 9	6 5	8 6	7 5	9 10	8 6	10 6	14 7	12 10	15 11	7 4	11 9	18 16
traffic light						10 8	3 2	4 5	12 10	4 3	11 9	10 9	15 13	12 8	10 8	11 8	17 12	8 7	9 7
vegetation							16 14	10 9	15 12	9 7	14 14	7 7	23 19	9 4	8 7	10 12	9 3	10 5	17 20
sky								2 0	6 6	13 10	7 4	12 8	12 8	17 13	8 4	6 4	15 13	24 23	8 5
fence									4 2	10 11	12 8	14 10	15 15	14 14	5 4	14 14	7 6	11 10	21 20
wall										6 4	3 2	11 8	14 10	15 13	3 2	8 5	14 12	8 6	13 10
terrain											6 4	11 8	16 13	25 19	5 1	16 13	17 15	19 17	16 14
person												18 14	21 22	4 3	7 3	10 7	3 2	16 11	7 6
rider													6 3	8 7	8 7	9 6	13 12	20 15	19 16
car														10 27	4 1	13 8	4 2	8 10	3 2
bicycle															10 10	11 9	1 0	17 16	14 11
bus																6 4	12 12	15 12	18 22
truck																	17 13	5 2	16 12
train																		28 27	19 14
motorcycle																			7 5

Table A.1: Analysis of pair of regions regarding the "Disconnected" spatial relationship on the CityScapes validation set. Table

Δ2 \ Δ1	road	sidewalk	building	pole	traffic sign	traffic light	vegetation	sky	fence	wall	terrain	person	rider	car	bicycle	bus	truck	train	motorcycle
road		5 3	6 4	16 4	2 2	3 1	4 5	1 0	7 5	5 2	4 2	14 5	10 4	7 4	5 5	2 2	5 3	4 3	12 8
sidewalk			3 3	33 26	3 1	1 0	8 9	3 2	4 2	6 3	12 9	35 22	25 13	22 14	13 11	6 5	7 3	7 5	9 6
building				30 24	16 12	5 4	18 13	10 3	15 11	10 8	7 4	8 3	9 12	24 23	18 15	10 8	9 7	23 18	5 4
pole					17 11	13 5	32 18	19 21	12 9	14 11	4 3	22 13	17 13	19 13	9 8	8 5	5 2	11 9	11 6
traffic sign						6 2	11 6	6 3	8 5	6 6	5 6	2 2	5 2	5 2	14 12	17 15	14 14	5 2	3 0
traffic light							11 9	7 3	5 5	13 10	8 5	4 3	2 1	5 7	7 4	10 9	13 11	7 5	3 1
vegetation								24 7	11 8	9 10	16 10	18 9	23 19	12 17	13 11	15 12	18 16	13 9	8 4
sky									3 1	4 3	4 1	5 3	11 12	4 4	8 10	21 17	25 22	4 4	4 1
fence										2 0	3 2	7 6	15 13	2 2	4 3	3 3	5 4	2 0	5 2
wall											3 3	9 4	6 6	4 3	6 4	7 8	9 10	8 3	6 5
terrain												13 10	14 11	9 8	12 10	4 3	16 13	3 5	3 2
person												15 11	21 18	16 5	14 10	19 13	21 20	10 7	15 12
rider													26 20	21 13	26 21	11 8	14 11	6 3	20 17
car														16 14	19 17	8 7	7 5	12 8	13 9
bicycle															27 20	3 2	2 0	10 7	14 11
bus																6 7	10 8	9 7	5 6
truck																	12 9	8 5	7 4
train																		0 0	4 3
motorcycle																			6 5

Table A.2: Analysis of pair of regions regarding the "Partially Overlapping" spatial relationship on the CityScapes validation set.

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — https://github.com/gallandarakhneorg/tex-templates
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

école doctorale **sciences pour l'ingénieur et microtechniques**

Title: Integration of Environment Contextual Knowledge in Deep Learning Modeling for Vision-based Scene Analysis

Keywords: Hybrid AI, Deep learning, Knowledge-Based systems, Visual tasks analysis

Abstract:

Computer vision has made an important evolution starting from traditional methods to advanced Deep Learning (DL) models. One of the goals of computer vision tasks is to effectively emulate human perception. The classical process of DL models is completely dependent on visual features, which only reflects how humans visually perceive their surroundings. However, for humans to comprehensively understand their environment, their reasoning not only depends on what they see but also on their pre-acquired knowledge. Addressing this gap is essential as achieving human-like reasoning requires a seamless combination of data-driven and knowledge-driven methods. In this thesis, we propose new approaches to improve the performance of DL models by integrating Knowledge-Based Systems (KBS) within Deep Neural Networks (DNNs). The goal is to empower these networks to make informed decisions by leveraging both visual features and knowledge to emulate human-like visual analysis. These methodologies involve two main axes. First, define the representation of KBS to incorporate useful information for a specific computer vision task. Second, investigate how to integrate this knowledge into DNNs to enhance their performance. To do so, we worked on two main contributions. The first work focuses on monocular depth estimation. Considering humans as an example, we can say that they can estimate their distance with respect to seen objects, even using just one eye, based on what is called monocular cues. Our contribution involves integrating these monocular cues as human-like reasoning for monocular depth estimation within DNNs. For this purpose, we investigate the possibility of directly integrating geometric and semantic information into the monocular depth estimation process. We suggest using an ontology model in a DL context to represent the environment as a structured set of concepts linked with semantic relationships. Monocular cues information is extracted through reasoning performed on the proposed ontology and is fed together with the RGB image in a multi-stream way into the DNNs for depth estimation. Our approach is validated and evaluated on widespread benchmark datasets. The second work focuses on panoptic segmentation task that aims to identify and analyze all objects captured in an image. More precisely, we propose a new informed deep learning approach that combines the strengths of DNNs with some additional knowledge about spatial relationships between objects. We have chosen spatial relationships knowledge for this task because it can provide useful cues for resolving ambiguities, distinguishing between overlapping or similar object instances, and capturing the holistic structure of the scene. More precisely, we propose a novel training methodology that integrates knowledge directly into the DNNs optimization process. Our approach includes a process for extracting and representing spatial relationships knowledge, which is incorporated into the training using a specially designed loss function. The performance of the proposed method was also evaluated on various challenging datasets. To validate the effectiveness of the proposed approaches for combining KBS and DNNs regarding different methodologies, we have chosen the urban environment and autonomous vehicles as our main use case application. This domain is particularly interesting because it is a challenging and novel field in continuous development, with significant implications for the safety, comfort and mobility of humans. As a conclusion, the proposed approaches validate that the integration of knowledge-driven and data-driven methods consistently leads to improved results. Integration improves the learning process for DNNs and enhances results of computer vision tasks, providing more accurate predictions. The challenge always lies in choosing the relevant knowledge for each task, representing it in the best structure to leverage meaningful information, and integrating it most optimally into the DNN architecture.

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France

école doctorale **sciences pour l'ingénieur et microtechniques**

Titre : Intégration de connaissances contextuelles dans des modèles à base d'apprentissage profond pour l'analyse de données visuelles

Mots-clés : IA hybride, Apprentissage profond, Systèmes à base de connaissance, Analyse de scènes par vision

Résumé :

La vision par ordinateur a connu une évolution importante, passant des méthodes traditionnelles aux modèles d'apprentissage profond. L'un des principaux objectifs des tâches de vision par ordinateur est d'émuler la perception humaine. En effet, le processus classique effectué par les modèles d'apprentissage profond dépend entièrement des caractéristiques visuelles, reflétant simplement la manière dont les humains perçoivent visuellement leur environnement. Cependant, pour que les humains comprennent l'environnement qui les entoure, leur raisonnement dépend non seulement de leurs capacités visuelles, mais aussi de leurs connaissances pré-acquises. Combler cette différence entre la perception humaine et celle des machines est essentielle afin de parvenir à un raisonnement similaire à celui des humains. Dans cette thèse, nous proposons de nouvelles approches pour améliorer les performances des modèles d'apprentissage profond en intégrant les systèmes basés sur les connaissances dans les réseaux de neuronaux profonds. L'objectif est d'aider ces réseaux à prendre les bonnes décisions en exploitant à la fois les caractéristiques visuelles et les connaissances pour émuler l'analyse visuelle de l'être humain. Ces méthodologies impliquent deux axes principaux. Premièrement, définir la représentation des connaissances pour incorporer des informations utiles à une tâche spécifique de vision. Deuxièmement, examiner comment intégrer ces connaissances dans les réseaux de neurones pour améliorer leurs performances. La première contribution porte sur l'estimation de la profondeur monoculaire. En effet, les humains sont capables d'estimer leur distance par rapport aux objets perçus, même en n'utilisant qu'un seul œil, et ceci en se basant sur les indices monoculaires. Nous proposons d'intégrer ces indices au sein des réseaux de neurones comme un raisonnement similaire à celui des humains pour l'estimation de la profondeur. À cette fin, nous suggérons d'exploiter un modèle ontologique pour représenter l'environnement comme un ensemble de concepts liés par des relations sémantiques. Les informations sur les indices monoculaires sont extraites grâce à un raisonnement effectué sur l'ontologie proposée et sont transférées dans les réseaux de neurones. Le deuxième travail porte sur la tâche de segmentation panoptique qui vise à identifier toutes les instances d'objets capturées dans une image. Nous proposons une approche qui combine les avantages des réseaux de neurones avec des connaissances sur les relations spatiales entre les objets. Nous avons choisi ce type de connaissances car elles peuvent fournir des indices utiles pour résoudre les ambiguïtés et distinguer entre les instances d'objets similaires. Plus précisément, nous proposons une stratégie d'entraînement qui intègre les connaissances dans le processus d'optimisation des réseaux de neurones. L'approche comprend un processus d'extraction et de représentation des connaissances sur les relations spatiales, qui sont incorporées dans l'entraînement sous forme d'une fonction de perte. Afin de valider l'efficacité des approches proposées, nous avons choisi l'environnement urbain et les véhicules autonomes comme principale cas d'application. Ce domaine est particulièrement intéressant car il s'agit d'un axe de recherche novateur en développement continu, avec des implications significatives pour la sécurité et la mobilité des humains. En conclusion, nous avons étudié diverses approches pour représenter les connaissances et les intégrer aux réseaux de neurones. Ces approches valident que l'utilisation combinée de méthodes basées sur les connaissances et celles basées sur les données conduit de manière constante à des résultats améliorés. Le défi principal réside toujours dans le choix des connaissances pertinentes pour chaque tâche, leur représentation et leur intégration de la manière la plus optimale dans l'architecture du réseau de neurones profond.

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France