Real-time Indoor Localization with Embedded Computer Vision and Deep Learning

Andrea Daou

Normandie Université

UNIVERSITÉ DE ROUEN NORMANDIE

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité **INFORMATIQUE**

Préparée au sein de l'**Université de Rouen Normandie**

Real-time Indoor Localization with Embedded Computer Vision and Deep Learning

Présentée et soutenue par

ANDREA DAOU

Thèse soutenue le 14/02/2024

devant le jury composé de :

MME SYLVIE CHAMBON	Maître de Conférences HDR - UNIVERSITE TOULOUSE 3 PAUL SABATIER	Rapporteur du jury
M. DRO DESIRE SIDEBE	Professeur des Universités - COMUE UNIVERSITES PARIS-SACLAY	Rapporteur du jury
M. FABRICE MERIAUDEAU	Professeur des Universités - UNIVERSITE DE BOURGOGNE	Président du jury
M. PAUL HONEINE	Professeur des Universités - Université de Rouen Normandie	Directeur de thèse
M. ABDELAZIZ BENSRHAIR	Professeur des Universités - INSA de Rouen Normandie	Co-directeur de thèse

Thèse dirigée par **PAUL HONEINE** (LABORATOIRE D'INFORMATIQUE DE TRAITEMENT DE L'INFORMATION ET DES SYSTEMES) et **ABDELAZIZ BENSRHAIR** (LABORATOIRE D'INFORMATIQUE DE TRAITEMENT DE L'INFORMATION ET DES SYSTEMES)

MI∫IΣ

litis

Acknowledgement

First and foremost, I would like to express my heartfelt gratitude to the DataHertz team who has provided me with a magnificent professional working environment that allowed me to develop my skills throughout this thesis. Special thanks are due to **Jean-Baptiste Pothin**, the head of the Research and Development department, whose warm welcome into his team and generous sharing of expertise have been crucial in my PhD journey. I will always be grateful for the time he has devoted to my thesis and the level of responsibility he has allowed me to take.

I would like to thank my supervisor **Paul Honeine**, whose constant leadership and availability have been the cornerstone of my PhD journey. His invaluable insights, constructive feedback, and unwavering support not only brought my research to a successful completion but also broadened and strengthened my scientific perspectives. I would also like to thank my co-supervisor **Abdelaziz Bensrheir**, whose remarkable research skills have continually guided me in the right direction. I am genuinely fortunate to have had such exceptional supervisors.

I would like to express my thanks to the Doctoral School of the University of Rouen Normandy and the LITIS laboratory for their assistance and resources throughout this thesis. I also thank the monitoring committee, **Fabrice Meriaudeau** and **Jean-Philippe Kotowicz**, that has followed me during these three years. The committee had an important positive impact on the quality of the work achieved in this thesis.

My deepest thanks go to **Fabrice Meriaudeau**, **Sylvie Chambon** and **Dro Désiré Sidibé** for the in-depth reading and evaluation of this thesis. Their thoughtful comments and suggestions have greatly contributed to the improvement of this work.

I would also like to take the opportunity to thank my friends and colleagues for their support during the last three years.

Finally and most importantly, I would like to thank my family, my father (**Andre**), my mother (**Guitta**), and my sister (**Lea**). I want to express my heartfelt gratitude to

every one of them, as they have been the fundamental pillars in shaping the person I am today. Their wise advice, empathetic listening, and unconditional love have played an instrumental role in my personal and academic growth. Throughout these years, their support and assistance have been the driving force that kept me moving forward, even during the most challenging and hard moments in my life.

Abstract

The need to determine the location of individuals or objects in indoor environments has become an essential requirement. The Global Navigation Satellite System, a predominant outdoor localization solution, encounters limitations when applied indoors due to signal reflections and attenuation caused by obstacles. To address this, various indoor localization solutions have been explored. Wireless-based indoor localization methods exploit wireless signals to determine a device's indoor location. However, signal interference, often caused by physical obstructions, reflections, and competing devices, can lead to inaccuracies in location estimation. Additionally, these methods require access points deployment, incurring associated costs and maintenance efforts. An alternative approach is dead reckoning, which estimates a user's movement using a device's inertial sensors. However, this method faces challenges related to sensor accuracy, user characteristics, and temporal drift. Other indoor localization techniques exploit magnetic fields generated by the Earth and metal structures. These techniques depend on the used devices and sensors as well as the user's surroundings.

The goal of this thesis is to provide an indoor localization system designed for professionals, such as firefighters, police officers, and lone workers, who require precise and robust positioning solutions in challenging indoor environments. In this thesis, we propose a vision-based indoor localization system that leverages recent advances in computer vision to determine the location of a person within indoor spaces. We develop a room-level indoor localization system based on Deep Learning (DL) and built-in smartphone sensors combining visual information with smartphone magnetic heading. To achieve localization, the user captures an image of the indoor surroundings using a smartphone, equipped with a camera, an accelerometer, and a magnetometer. The captured image is then processed using our proposed multiple direction-driven Convolutional Neural Networks to accurately predict the specific indoor room. The proposed system requires minimal infrastruc-

ture and provides accurate localization. In addition, we highlight the importance of ongoing maintenance of the vision-based indoor localization system. This system necessitates regular maintenance to adapt to changing indoor environments, particularly when new rooms have to be integrated into the existing localization framework. Class-Incremental Learning (Class-IL) is a computer vision approach that allows deep neural networks to incorporate new classes over time without forgetting the knowledge previously learned. In the context of vision-based indoor localization, this concept must be applied to accommodate new rooms. The selection of representative samples is essential to control memory limits, avoid forgetting, and retain knowledge from previous classes. We develop a coherence-based sample selection method for Class-IL, bringing forward the advantages of the coherence measure to a DL framework. The relevance of the methodology and algorithmic contributions of this thesis is rigorously tested and validated through comprehensive experimentation and evaluations on real datasets.

Résumé

La localisation d'une personne ou d'un bien dans des environnements intérieurs est devenue une nécessité. Le système de positionnement par satellites, une solution prédominante pour la localisation en extérieur, rencontre des limites lorsqu'il est appliqué en intérieur en raison de la réflexion des signaux et de l'atténuation causée par les obstacles. Pour y remédier, diverses solutions de localisation en intérieur ont été étudiées. Les méthodes de localisation en intérieur sans fil exploitent les signaux pour déterminer la position d'un appareil dans un environnement intérieur. Cependant, l'interférence des signaux, souvent causée par des obstacles physiques, des réflexions et des appareils concurrents, peut entraîner des imprécisions dans l'estimation de la position. De plus, ces méthodes nécessitent le déploiement d'infrastructures, ce qui entraîne des coûts d'installation et de maintenance. Une autre approche consiste à estimer le mouvement de l'utilisateur à l'aide des capteurs inertiels de l'appareil. Toutefois, cette méthode se heurte à des difficultés liées à la précision des capteurs, aux caractéristiques de mouvement de l'utilisateur et à la dérive temporelle. D'autres techniques de localisation en intérieur exploitent les champs magnétiques générés par la Terre et les structures métalliques. Ces techniques dépendent des appareils et des capteurs utilisés ainsi que de l'environnement dans lequel se situe l'utilisateur.

L'objectif de cette thèse est de réaliser un système de localisation en intérieur conçu pour les professionnels, tels que les pompiers, les officiers de police et les travailleurs isolés, qui ont besoin de solutions de positionnement précises et robustes dans des environnements intérieurs complexes. Dans cette thèse, nous proposons un système de localisation en intérieur qui exploite les récentes avancées en vision par ordinateur pour localiser une personne à l'intérieur d'un bâtiment. Nous développons un système de localisation au niveau de la pièce. Ce système est basé sur l'apprentissage profond et les capteurs intégrés dans le smartphone, combinant ainsi les informations visuelles avec le cap magnétique du

smartphone. Pour se localiser, l'utilisateur capture une image de l'environnement intérieur à l'aide d'un smartphone équipé d'une caméra, d'un accéléromètre et d'un magnétomètre. L'image capturée est ensuite traitée par notre système composé de plusieurs réseaux neuronaux convolutionnels directionnels pour identifier la pièce spécifique dans laquelle se situe l'utilisateur. Le système proposé nécessite une infrastructure minimale et fournit une localisation précise. Nous soulignons l'importance de la maintenance continue du système de localisation en intérieur par vision. Ce système nécessite une maintenance régulière afin de s'adapter à l'évolution des environnements intérieurs, en particulier lorsque de nouvelles pièces doivent être intégrées dans le système de localisation existant. L'apprentissage incrémental par classe est une approche de vision par ordinateur qui permet aux réseaux neuronaux profonds d'intégrer de nouvelles classes au fil du temps sans oublier les connaissances déjà acquises. Dans le contexte de la localisation en intérieur par vision, ce concept doit être appliqué pour prendre en compte de nouvelles pièces. La sélection d'échantillons représentatifs est essentielle pour contrôler les limites de la mémoire, éviter l'oubli et conserver les connaissances des classes déjà apprises. Nous développons une méthode de sélection d'échantillons basée sur la cohérence pour l'apprentissage incrémental par classe dans le cadre de l'apprentissage profond. La pertinence de la méthodologie et des contributions algorithmiques de cette thèse est rigoureusement testée et validée par des expérimentations et des évaluations complètes sur des données réelles.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

“ Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less. ”

Marie Curie

Sommaire

Indoor localization systems have piqued the interest of both academia and industry because of their numerous applications, ranging from security and military use to monitoring and tracking in homes, commercial buildings, airports, hospitals, and university campuses. The Global Navigation Satellite System (GNSS) is a leading solution for outdoor localization. Yet, its effectiveness diminishes indoors due to numerous multi-path reflections and considerable attenuation of signals caused by obstacles such as walls. To solve this issue, various indoor localization systems are being created. Vision-based indoor localization systems are a category of indoor localization solutions that use computer vision and image processing techniques to

detect the location or the position of people or objects within indoor environments.

1.1 Thesis Collaboration

This CIFRE (*Convention Industrielle de Formation par la Recherche*) PhD thesis is part of a long and fruitful collaboration between the LITIS (*Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes*) laboratory and DataHertz company [KPHB18, KPHB19].

The LITIS laboratory, founded in 2006, is a research unit in information science and technology affiliated with the University of Rouen Normandy, the University of Le Havre Normandy, and the National Institute of Applied Sciences of Rouen Normandy. The LITIS is a founding member of the CNRS (*Centre National de la Recherche Scientifique*) research federation NormaSTIC. The laboratory's methodology is distinctly multidisciplinary, fostering collaborations between both practitioners and theoreticians at the intersection of computer science, artificial intelligence, signal and image processing, and applied mathematics. The laboratory projects and applications reach several industries, including the development of intelligent mobility systems, the processing of medical data, and the protection of cultural assets. The LITIS is divided into seven different teams, notably including the APP (*Apprentissage*) and STI (*Systèmes de Transport Intelligents*) teams.

The APP team specializes in machine learning, using statistical modeling and learning methodologies to extract knowledge from data of various nature (*e.g.*, signal, image, and graph). The research applications encompass handwriting recognition, ancient and historical document analysis, information retrieval, medical imaging, time series analysis, as well as scene analysis and description. On the other hand, the STI team is committed to leveraging information science and technology for Advanced Driving Assistance Systems (ADAS) by focusing on computer vision. The research of this team focuses on the design of onboard autonomous systems capable of delivering valuable real-time information to drivers and passengers, especially in resource-constrained environments with limited infrastructure and degraded conditions. The goal of this team research is to enhance transport optimization, risk prevention, and safety. This thesis project aligns with both of these teams.

DataHertz company, founded in 2010, is an innovative Small and Medium-sized Enterprise (SME) with a unique expertise in communication networks. DataHertz specializes in the design, development, integration, and maintenance of private

mobile radio and wireless broadband installations. Its main customers are public institutions and companies as well as industrial groups. The DataHertz teams are present throughout France, offering high-quality services and developing communication networks adapted to the needs of diverse industries and professions in the most extreme conditions. DataHertz's Research and Development (R&D) activity offers innovative and secure technologies for voice, video, and high-speed data transmission over fixed or mobile networks. The R&D team, based in Troyes, includes engineers, PhD researchers, and interns with the following responsibilities:

- Design and definition of product evolution.
- Maintenance and support for current products.
- Development, testing, and validation of future products.
- Custom development and prototyping for critical projects.

This thesis project is seamlessly integrated into the suite of indoor/outdoor localization solutions provided by DataHertz. The "Polyalerte" localization solution, developed by DataHertz, has been rigorously crafted to correspond with the different objectives expressed by customers across a range of industries (*e.g.,* prisons, security companies, industrial sites, municipal police forces, university campuses, etc.). This indoor/outdoor localization solution uses Global Positioning System (GPS), High Frequency (HF) beacons or Radio Frequency Identification (RFID) tags to improve the safety and security of both assets and humans. Our research activities aim to develop and expand the capabilities of DataHertz company's localization products, specifically tailored to the demands of firefighters, police officers, military and defense personnel, as well as lone workers, to ensure safety, security, and efficiency in various operational scenarios.

1.2 Research Context

Localization and positioning services are gaining popularity around the world as people want reliable location-based information for a variety of applications, such as navigation, monitoring, tracking, and information services [FHS18, KKAMAA20, AAW+20]. To meet this requirement, a wide range of strategies based on diverse technologies have arisen to provide precise positioning solutions in both outdoor

and indoor situations. In recent years, the expansion of mobile devices and their applications has resulted in the introduction of new services for users. This surge in mobile technology adoption has further fueled the demand for localization and positioning services [AHP20]. Indoor localization solutions are a collection of technologies, systems, or methods for determining and tracking the location of people or objects within an indoor environment, often a building or an enclosed space. Indoor localization solutions serve several key industries, mainly:

- Emergency services: Security stakeholders such as firefighters, security guards, police officers, and military personnel require precise and robust positioning systems during critical operations.
- Lone workers protection and safety: Workers operating in remote or isolated environments demand vital safety solutions. This is common in various industries, including transportation, tertiary sectors (*e.g.,* hotels, cultural and sports facilities), as well as municipal services.
- Marketing and management: Mobile devices users require positioning systems to enhance their overall navigation experience. These systems allow them to access route guidance and to discover nearby businesses or shops. Additionally, owners and operators of large enclosed spaces such as airports, museums, and shopping malls demand positioning solutions to improve their services and security.

The GNSS (*e.g.,* GPS and Galileo) is a dominant solution for outdoor localization based on trilateration, a process in which a GNSS mobile device receiver's position is identified based on signals received from the GNSS satellites orbiting above. A minimum of four separate satellites are necessary. Each of these satellites emits a signal including both its precise location and the current time. The mobile device GNSS receiver analyzes these signals and performs the necessary calculations to estimate its distance from each satellite and therefore its position. The delay in signal exchange between a mobile device and a satellite determines the distance between them [Ble97]. However, the performance of GNSS signals deteriorates inside buildings due to numerous multi-path reflections and significant signal attenuation through walls. To address this issue, localization systems based on other technologies are being developed [KKAMAA20].

Wireless-based indoor localization technologies include a variety of methods and systems that rely on wireless signals to determine a device's location in indoor

environments such as infrared [HK10, AL20], ultrasound [MAG+02, QL17], and RFID tags [CWG+20]. Furthermore, the increasing number of wireless networks has encouraged the development of indoor localization systems based on technologies such as Ultra-Wide-Band (UWB) [SM16], WiFi [ZGL19, SFH+19, LLY+20], or Bluetooth [TLG+20, MSG+20]. While these technologies offer promising solutions for indoor localization, they share several challenges and limitations. These systems are vulnerable to signal interference, often resulting from physical obstructions, reflections and competing devices, leading to inaccuracies in location estimation. In addition to this, these technologies necessitate reference points with predetermined coordinates, known as base stations, tags, access points, or beacons, depending on their function within the localization system. Thus, their deployment needs the installation and maintenance of a transmitter/receiver infrastructure, which incurs high costs and complex logistics [ACHC20, ACH18]. Another approach for indoor localization is the use of dead reckoning techniques which removes the restriction of dependence on reference points. Pedestrian Dead Reckoning (PDR) estimates user's movement based on accelerometer and magnetometer sensors (and/or gyroscope) commonly found in devices like smartphones [KH14, KPC15, AHP18, KNZC18]. The primary challenge with such approaches lies in calculating distance and orientation based on the measured signals, which are susceptible to numerous factors such as sensor placement and accuracy as well as individual's physical characteristics and activities. Another issue is temporal drift, which results from the accumulation of localization errors over time. Indoor localization techniques that use magnetic fields are also proposed [AHP20, TAC+21, OAM22]. These systems use the Earth's magnetic field for localization, taking advantage of anomalies caused by metal structures in indoor environments. The difficulties in such a method come from the dependence of the magnetic field measures on the used devices and sensors as well as on the users' surroundings.

In the current state of the art, vision-based localization has gained prominence. This shift has been made possible by the progress of computer vision algorithms and the widespread availability of camera-equipped devices. Vision-based techniques hold considerable appeal due to their independence from radio and wireless signal measurements, as they rely on the visual data captured by the camera, and the nonessential infrastructure installation and its maintenance. Vision-based methods offer a proficient and precise positioning solution. Indoor environments are relatively complicated because of layout variability, object and decoration complex-

ity, multi-scale and viewpoint changes, as well as lighting conditions. Vision-based robust algorithms try to mitigate indoor environmental variations and complexity.

Traditionally, vision-based indoor localization heavily relied on handcrafted feature extraction techniques, like Scale-Invariant Feature Transform (SIFT) [Lin12] and Speeded Up Robust Features (SURF) [BETVG08], which allow for the extraction of real or integer characteristics, as well as Oriented FAST and rotated BRIEF (ORB) [RRKB11] and Binary Robust Invariant Scalable Keypoints (BRISK) [LCS11], which allow for the extraction of binary characteristics. Handcrafted feature extraction techniques entail the manual design and extraction of distinctive visual features from images collected within indoor environments. These features are intended to incorporate important visual information, making them appropriate for tasks such as scene recognition. During the offline phase, features are extracted from images taken by smartphones or other devices cameras within the indoor environment using a handcrafted feature extraction technique. Then, a dataset with features representing each indoor location is created and prepared. These features capture unique patterns, shapes, or textures present in the indoor scenes. During the online phase, the system matches the query image features with the previously collected features dataset using traditional classifiers, such as Support Vector Machines (SVM) [LW18], K-Nearest Neighbors (KNN) [Pet09], or Random Forests [Bre01]. This allows the determination of the user device's location. These traditional techniques necessitate expertise in feature design and selection, and they frequently entail complex algorithms to efficiently identify and match features, which often struggle to adapt to the complexity of indoor environments. Additionally, given that extracting relevant local features from an image and searching for correspondences in a large mass of data is time-consuming, the real-time processing constraint reduces the use of these conventional methods.

Previous work, based on a collaboration between the LITIS laboratory and Data-Hertz company, focused on using SIFT local features for localization in known environments [KPHB18, KPHB19]. However, this research showed some limits in both feature indexing and correspondence search. It also highlighted restrictions with computational complexity and memory use. This PhD thesis aims to overcome these limitations, by taking full advantage of recent advances in machine learning for computer vision with Deep Learning (DL).

With the emergence of DL, the landscape of computer vision has undergone a paradigm shift [LBH15, ZYT17]. This revolutionary technology, distinguished by

deep neural networks architectures, has changed the way we approach and perceive various problems. This thesis aims to overcome difficulties in indoor localization, thanks to recent advances in DL and recent progress in reducing computational complexity and memory space.

1.3 Research Purpose

This thesis addresses the problem of localization in indoor human environments. GNSS-based localization is not feasible in such environments, and the proposed wireless-based alternatives can incur significant costs for infrastructure deployment and maintenance. The main aim of this thesis is to develop a system that allows a user to locate himself in real-time inside a building, using an embedded camera (*e.g.,* smartphone or tablet), or an augmented reality glasses (*e.g.,* Google Glass). The chosen approach consists in locating the camera's carrier based on the images he captures of his surroundings.

In this thesis, we provide new solutions and improvements to existing methods for the complex problem of indoor scene recognition using pre-existing technologies embedded in smartphones, which are accessible across various brands and models. Indoor scene recognition, considered a key component for vision-based localization systems, is the process of identifying an indoor scene from an indoor environment based on visual data (typically images or video frames). The main focus of this thesis is the use of a conventional monocular camera for indoor scene image acquisition for reasons of cost, practicality, and computation time. These cameras commonly available in end-user devices, such as smartphones, capture imagery in the form of RGB images, making them practical and accessible for indoor localization scenarios. We aim to extract valuable visual cues and features from the images describing the indoor environments using computer vision and DL methods to aid in localization tasks. Additionally, our research is grounded in the context of known indoor environments, where datasets describing the indoor space have been carefully acquired and maintained. Having prior knowledge of the indoor environments allows us to leverage this information during the localization process. To enhance performance, the system will also exploit other sensors found in the user's end-device. Our system does not require costly implementation and subsequent maintenance costs, making it a cost-effective and hassle-free solution for indoor localization.

Vision-based indoor localization faces several challenges, all of which add to the task's complexity. The inherent intra-class variability and inter-class similarity within indoor scenes necessitate advanced scene recognition algorithms. The dynamic nature of indoor environments, influenced by factors such as varying light conditions as well as moving furnishings and humans, adds another layer of complexity. Furthermore, occlusion limits visibility and complicates accurate localization. Additionally, indoor environments are subject to scene modifications and additions that necessitates adaptability in localization systems to maintain efficacy over time.

Implementing vision-based indoor localization on embedded systems encounters inherent limitations. Possessing millions and billions of parameters, deep neural networks are naturally greedy for computing power and memory, making them difficult to use and deploy on embedded systems [NJKN20]. Indeed, this thesis is aimed at implementation on smartphones, thus requiring solutions with low power consumption and low memory capacity. Reducing the number of parameters and computational requirements, while preserving performance, is very important for the deployment of deep neural networks. Real-time constraint is also a critical consideration for indoor localization systems embedded in smartphones. To meet the real-time processing requirements, the time taken for processing an image should be less than the acquisition time between two images. This temporal restriction poses challenges for traditional computer vision methods. In this context, deep neural networks prove to be more suitable [AZH+21].

Throughout this study, we are operating within the scenario of identifying individual rooms. This means that the primary objective of our research is to develop a system that can accurately determine and distinguish different rooms within a building or enclosed space. The accuracy of localization at the level of entire rooms within a building is provided by room-level solutions. They concentrate on figuring out what room a user is in. It can be applied to smart buildings to track assets [TKVT18], keep an eye on room occupancy [JJCS15], provide general location awareness [GPMSSA21], etc. This room-level vision-based indoor localization system is suitable for applications where coarse location information is sufficient. It is designed for people who need positioning solutions that are accurate, robust, and less susceptible to conditions that alter waves in indoor environments like firefighters, police officers, lone workers, visually impaired people, dependent elderly people, office buildings and university campus visitors, as well as airport travelers.

In this thesis, we also highlight the importance of ongoing improvement and maintenance of vision-based indoor localization systems. This system requires maintenance to adapt to evolving indoor environments, especially when new rooms need to be integrated into the existing indoor localization system. This is where Class-Incremental Learning (Class-IL) must be applied. Class-IL is a computer vision approach that enables deep neural networks to incorporate new classes over time without forgetting the knowledge of previously learned classes. In the context of vision-based indoor localization, this approach is essential for incrementally updating the localization model, facilitating its seamless adaptation to new rooms as well as controlling memory and computation limitations.

1.4 General Contributions

In this thesis, our main contributions are as follows:

- **Introducing a direction-driven multiple Convolutional Neural Networks (CNNs) system for indoor localization:** This system is based on a combination of image features and the magnetic heading from a smartphone. The proposed architecture is composed of four CNNs, each specific to a definite heading range, trained on a dataset containing images with their respective magnetic heading.

- **Proposing a hybrid "server and on-device" computing approach:** This method addresses latency, scalability, and privacy challenges in indoor localization systems while meeting the computational requirements of DL and respecting the end-user devices' limitations.

- **Providing a novel indoor scenes dataset:** The created dataset contains real images with their respective magnetic heading direction in the metadata. While there are various indoor and/or outdoor localization datasets in the literature, none of them incorporate data other than images.

- **Introducing a coherence-based sample selection strategy for Class-IL:** This sampling method is based on the coherence measure to maximize the exemplars diversity for Class-IL. This is the first time that the coherence measure is investigated for DL, and more specially for Class-IL. Class-IL is critical for maintaining and updating a vision-based indoor localization system as new

rooms need to be integrated into the existing indoor localization system with time.

1.5 Thesis Outline

The rest of the manuscript is divided into four main chapters. The first two chapters are dedicated to the state of the art of indoor localization technologies and techniques, along with the application of smartphone sensors in indoor localization. The following chapter concerns the development of our indoor localization approach based on smartphone's camera, accelerometer, and magnetometer sensors. The last chapter focuses on coherence-based sampling for Class-IL, shedding light on the importance of maintaining and updating a vision-based indoor localization system. In the following, we present a brief overview of the manuscript's chapters.

Chapter 2: Fundamentals of Vision-based Indoor Localization

This chapter introduces indoor localization technologies and techniques. It briefly reviews several technologies, setting the stage for an in-depth examination of vision-based indoor localization, with a focus on scene recognition. Additionally, it explores the area of DL in the context of indoor scene recognition, more specially CNNs. These deep neural networks revolutionized image classification by enabling automated and accurate recognition of complex visual patterns. Within this scope, this chapter examines existing approaches, datasets, and challenges that demand careful consideration.

Chapter 3: Smartphones in Indoor Localization Systems

This chapter delves into the application of smartphone built-in sensors in indoor localization. It includes an overview of the capabilities and constraints of these sensors when employed for location estimation as well as some existing smartphone-based indoor localization systems. We also present the DL frameworks interoperability for smartphone deployment. In addition, this chapter provides a thorough examination of the advantages and disadvantages of current computing approaches in this context.

Chapter 4: Multi-sensor Data Fusion for Indoor Localization

This chapter presents an approach that takes advantage of smartphone sensors

combined with computer vision, more specially DL, for indoor room-level localization. As smartphones are not only endowed with cameras but also equipped with several other sensors, such as accelerometers and magnetometers, they provide the opportunity to acquire additional information and therefore build reliable localization systems. We propose a novel direction-driven multi-CNN indoor scene recognition system based on image features and the magnetic heading from a smartphone camera. Results on a real dataset show that the proposed method outperforms the scene recognition method based solely on image features in terms of accuracy. To address latency, scalability, and privacy issues, a hybrid computing strategy is also presented.

Chapter 5: Coherence-based Sample Selection for Class-incremental Learning

This chapter presents a novel sample selection strategy for Class-IL. In order to handle memory constraints and prevent forgetting previously acquired knowledge, the careful selection of representative samples is important. The proposed approach is based on the coherence measure, which was originally used with linear and kernel-based models. This chapter investigates the coherence measure for diverse samples selection in a DL context. The coherence-based sample selection method is validated on two well-known datasets. The obtained results show that the proposed method outperforms state of the art techniques, with better average test accuracy. This chapter sheds light on the importance of Class-IL for maintaining a vision-based indoor localization system in changing environments over time, especially when adding new rooms to an existing indoor localization system.

Finally, a general conclusion assesses the results of this thesis and suggests a number of future research directions.

1.6 Research Publications

Peer-reviewed international journal paper (1 + 1)

- A. Daou, J.B. Pothin, P. Honeine, and A. Bensrhair. Indoor scene recognition mechanism based on direction-driven convolutional neural networks. Sensors, 23(12):110439, 2023.

- A. Daou, J.B. Pothin, P. Honeine, and A. Bensrhair. Coherence-based sample selection for class-incremental learning. Pattern Recognition Letters. Submitted on October 2023.

Peer-reviewed national conference paper (1)

- A. Daou, J.B. Pothin, P. Honeine, and A. Bensrhair. Contrôle d'un système multi-cnn via le cap magnétique du smartphone pour la reconnaissance de scènes indoor. In Actes du 28-ème Colloque GRETSI sur le Traitement du Signal et des Images, Nancy, France, 6 - 9 September 2022.

Workshop with proceedings (1)

- A. Daou, J.B. Pothin, P. Honeine, and A. Bensrhair. Amélioration des performances des réseaux de neurones convolutifs en localisation indoor par augmentation des données. In Actes de la 18-ème édition d'ORASIS, Lac de Saint-Ferréol, France, 13 - 17 September 2021.

Chapter 2

Fundamentals of Vision-based Indoor Localization

Sommaire

2.1 Introduction

Indoor location-based services constitute an important part in our daily life, providing position and direction information of a person or an object in an indoor

space. These systems can be useful in security and monitoring applications that target specific areas such as rooms. Indoor localization has profound implications across a range of applications such as navigation and tracking [KKAMAA20], retail and marketing [HKLK17, HQY+22], robotics [HJLT23] and many more. According to Technavio [1], the global Indoor Positioning and Indoor Navigation (IPIN) market is expected to increase by USD 52,503.46 million between 2022 and 2027, with a Compound Annual Growth Rate (CAGR) [2] of 34.07%. Different strategies have been developed with varying degrees of success to address the needs of localization. Indoor and outdoor localization challenges differ significantly due to the unique characteristics of each environment. GNSS is the most well-known and frequently used system in modern times. Unfortunately, GNSS only works in outdoor environments and fails in indoor environments because the satellite network requires visibility.

In an indoor context, alternatives such as UWB, RFID tags, and more recently Bluetooth and WiFi have been proposed [ZGL19]. These new solutions require a transmitter/receiver infrastructure before implementing the localization application. Considering the deployment requirement and the maintenance costs they may incur, computer vision-based approaches prove to be a highly promising solution. These approaches require (almost) no infrastructure, and it is possible to achieve more accurate localization than some of the aforementioned strategies [MPS14].

Scene recognition forms the foundation of vision-based indoor localization, which is defined as the task of identifying accurately a room from a given image. In scene recognition, the goal is to classify an input image into a predefined set of scene classes. As the system analyzes the visual information in the image, several discriminating characteristics play a crucial role in the recognition process, mainly: environment data, sensing devices, detected elements, and the used localization method. Depending on the configuration, the number of cameras, and the nature of the environment, the challenges can be more or less complex [MMM+20]. Despite years of research in this field, indoor scene recognition remains an open problem due to the different and complex places in the real-world. Indoor environments are relatively complicated because of layout variability, objects and decorations complexity, multi-scale and viewpoint changes.

In this chapter, we present a comprehensive exploration of indoor localization

[1] Technavio; Website: `https://www.technavio.com/report/indoor-positioning-and-indoor-navigation-market-industry-analysis`, accessed on 12 September 2023.

[2] CAGR is the annual growth of a person's investments over a specific period of time.

techniques, starting with an overview of several technologies. Then, we focus on vision-based indoor localization in known environments, covering essential topics like scene understanding and recognition, allowing systems to interpret and navigate complex indoor environments using visual cues. Furthermore, we delve into the field of DL for vision-based indoor localization, specially concentrating on CNNs which transformed image classification, making it possible to automatically and precisely identify intricate visual features. In this context, we investigate existing approaches, datasets, and the challenges that must be addressed.

2.2 Background on Indoor Localization Techniques

People all over the world are increasingly interested in localization and positioning services. Location-based services, relying on location data, provide precise and real-time information on a user's or an object's location in a given environment. Several techniques based on different technologies are available to provide an accurate positioning solution in outdoor and indoor environments [AM22]. Accurate localization is critical in many domains, providing numerous benefits and enabling a wide range of applications in different domains like navigation systems [AAW+20], augmented reality [JS23], asset tracking [FHS18], and emergency response [FFCM17]. Indoor localization systems aid users in navigating through complex indoor spaces such as shopping malls, airports, hospitals, and large office complexes, making it easier for them to locate specific points of interest, such as stores, gates, or meeting rooms, which improves the overall user experience.

While GPS [KKAMAA20] and Point Of Interest (POI) [LTC21] have been widely used for outdoor localization, indoor localization presents unique challenges due to low signal strength and reduced accuracy in enclosed and cluttered environments. Indoor localization systems are classified according to their sensing technologies and measurement methods. The sensing technologies refer to the sensors utilized, while the measurements techniques refer to the methods and metrics used in localization. There are two main approaches to the indoor localization problem: infrastructure-based systems, which require a transmitter/receiver infrastructure, and infrastructure-free systems, which can operate (almost) autonomously [AHP18]. See Figure 2.1 for the different indoor localization techniques.

The diverse nature of indoor localization solutions, the different elements that influence their performance, and the multiple applications they serve present a

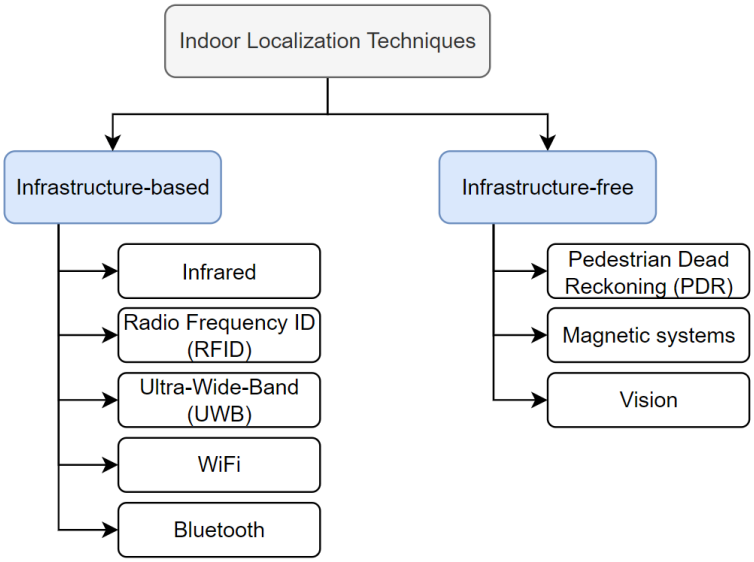


Figure 2.1: Indoor localization techniques.

number of problems for the design and the development of a methodology that delivers meaningful results. To be considered effective and practical, an indoor localization system must meet a number of criteria and needs. Indoor localization solutions are characterized by several elements [GFM21, SHK22]. See Figure 2.2 for the various elements representing an indoor localization solution.

The characteristics of an indoor localization system depend on a variety of factors, such as the utilized agents that encompass devices (*e.g.*, RFID tags, WiFi access points, Bluetooth beacons, smartphones, etc.) and sensors (*e.g.*, camera, accelerometer, magnetometer, barometer, etc.). These devices and sensors have a pivotal role in gathering essential data and processing it. Furthermore, the underlying technologies and applied techniques (see Sections 2.2.1 and 2.2.2), which include used methods and algorithms (*e.g.*, fingerprinting, triangulation, trilateration, sensor fusion, machine learning, computer vision, etc.), represent a key element of the solution. Integrating several technologies and sensors can be complex. As a result, a methodology should consider how various technologies can coexist to produce accurate and consistent results.

The studied environment has an impact on the solution as well. The application environment might range from office buildings, airports, hospitals to warehouses. Indoor localization solutions need to be customized for individual use cases and modifications must be applied in order to effectively meet the application requirements. In indoor localization systems, place representation entails gathering and

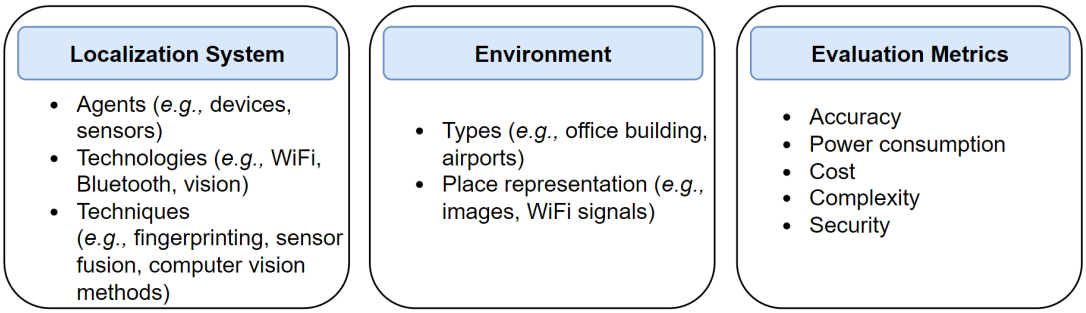


Figure 2.2: Elements of an indoor localization solution.

encoding information about the indoor environment studied. This representation includes a variety of data types, such as images, WiFi signal strengths, magnetic fingerprints, or other sensors information.

Additionally, the choice of evaluation metrics depends on the unique goals and needs of the indoor localization solution, such as accuracy, complexity, security, power consumption, cost, scalability, latency and many more. The achievement of high accuracy, in line with users' expectations, is a crucial necessity for localization systems. The user experience is also greatly influenced by energy efficiency, which emphasizes the necessity for optimum energy utilization to prolong device battery life. The difficulty of cost effectiveness also looms big. Utilizing existing infrastructure is a crucial cost-saving approach since systems that depend on new infrastructure frequently have higher costs than those that do not. Another crucial element is scalability, which ensures adaptation to varied user densities dispersed over large areas. The key to creating a complete localization system is striking a balance between a variety of evaluation metrics. A careful consideration of evaluation metrics is essential for designing, optimizing, and validating indoor localization solutions. Thorough validation is required for meaningful results like testing real-world scenarios and recreating difficult conditions.

2.2.1 Infrastructure-based Techniques

The majority of systems with infrastructure that have been popular for indoor localization use wireless technologies such as infrared [HK10, AL20], UWB [SM16], WiFi [SFH+19, LLY+20], Bluetooth [TLG+20, MSG+20], RFID tags [CWG+20], and sensor fusion techniques that combine several wireless technologies [TC22]. Diverse sensor technologies used for indoor localization are represented in Figure 2.3.

Figure 2.3: Examples of sensor technologies for indoor localization.

Wireless signals propagate differently indoors than outdoors due to issues such as signal attenuation, multi-path interference, and signal reflections [LLY+20]. This involves the development of specialized localization algorithms and techniques. In wireless-based indoor localization systems, various techniques such as fingerprinting [YWW20], trilateration [Shc14], and Received Signal Strength Indicator (RSSI) measurements [SS18] are used. Fingerprints involve creating a database of signal characteristics at known locations and matching them to the received signals to estimate the user's position. Trilateration relies on measuring the distances between the user and multiple transmitters/receivers. RSSI-based methods estimate the user's position based on the Received Signal Strength (RSS).

Wireless-based indoor localization systems confront a number of challenges that can reduce their applicability and accuracy. The deployment of transmitters and receivers, as well as the calibration of the system, necessitate careful planning and configuration, resulting in high installation expenses. Additionally, maintaining that infrastructure can be costly. These systems are also susceptible to environmental conditions that alter the waves, such as walls, furniture, and human beings [ACH18, ACHC20].

2.2.2 Infrastructure-free Techniques

Continuous studies seek to improve the accuracy, reliability, and scalability of wireless-based indoor localization systems. This includes creating advanced signal processing algorithms, computer vision techniques, and hybrid systems that use other sensor modalities, such as initial sensors and cameras.

Indoor localization techniques encompass diverse methods, including naviga-

tion by estimation (*e.g.,* PDR) based on acceleration, gyroscope and magnetic sensors, which estimate movement by continuously updating the device's estimated position based on its initial known position and the collected sensor data [KH14, AHP18, KNZC18]. Challenges involve accurately extracting distance and orientation from these signals amidst factors like sensor accuracy and user activities. Navigation by estimation frequently incorporates additional data sources or sensor fusion techniques to improve accuracy. This might include periodic correction using known reference points or incorporating external data like WiFi or Bluetooth beacons [SLK18, PEH19, RK22].

Magnetic field has also been used for localization and tracking in a variety of applications [AHP20, TAC+21, OAM22]. These indoor localization systems rely on the Earth's magnetic field and leverage anomalies created by metal structures within the indoor environments, eliminating the need for an infrastructure installation. The magnetic field of the Earth is a natural phenomena. It maintains a consistent magnetic field strength across vast distances and does not undergo sudden changes over short shifts like a few meters. However, the presence of ferromagnetic materials within indoor environments introduces anomalies [ZWWN15, SGD13]. These anomalies, detectable by a magnetometer, can serve as fingerprints, enabling location estimation. However, these systems face challenges due to the variability in building structures, interference from electronic devices, and the need for careful calibration.

In the current landscape, vision-based localization emerges as a prominent trend, leveraging advancements in computer vision algorithms and camera-equipped devices. These systems create databases of images labelled with geographic information and then use these databases to match and locate new images during online localization. Vision-based methods do not rely on radio/wireless signals and offer efficient and accurate positioning. While environmental variations exist due to factors like viewpoint, light conditions, and many more limitations, robust algorithms aim to mitigate these effects. An alternative approach employs Simultaneous Localization and Mapping (SLAM) techniques [LRH+19, KRM+21], which, unlike prior methods, construct environment maps alongside localization, employing methodologies like Kalman filters and Expectation Maximization Algorithms.

Vision-based techniques are of great interest because they do not require implementing and maintaining an infrastructure, unlike other indoor technologies.

Choosing a suitable solution is dependent not only on the application domain, but also on the application's specific needs such as accuracy, computing time, memory size, and equipment.

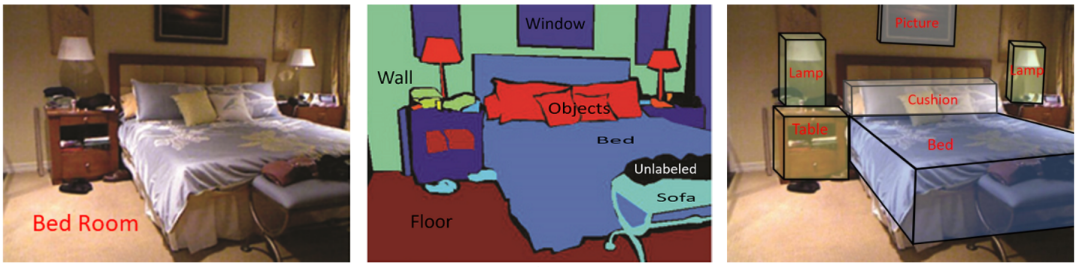
2.3 Vision-based Indoor Localization in Known Environments

Vision-based localization aims to answer the fundamental question "Where am I?" by leveraging computer vision techniques and image analysis. The system seeks to determine the location of a user within an environment by recognizing the specific characteristics and scene category of that location. In this section, we will go into a comprehensive exploration of scene recognition in the context of localization. Our focus will be directed to the complex area of indoor scene recognition.

2.3.1 Scene Recognition for Localization

In the context of localization, vision-based methods have shown great interest in recent years [Hu15]. Indeed, tracking methods based on visual image recognition (*i.e.*, image features) require only an image acquisition device. The cognitive ability of people to perceive and analyze the visual aspects and elements present in an environment is referred to as scene understanding. It includes tasks such as scene recognition, semantic segmentation, and object recognition and detection [NKP18] (see Figure 2.4). Scene understanding attempts to grasp the underlying meaning and context of a scene by using the visual information collected in images serving as the framework for higher-level analysis and interpretation.

Scene recognition is concerned with categorizing the entire scene, taking into account the general structure, context, and functionality of the indoor or the outdoor environment [TSYW17]. Scene recognition can be described as classifying a scene query image to one many scene categories, serving as the bridge between raw visual data and meaningful spatial information. Traditional vision-based methods for scene recognition mainly focus on images features, which include the image's global content, objects, and layout visual comprehension. The scene recognition system must therefore have a thorough understanding of the scenes that we encounter in daily life, including both indoor and outdoor environments, in order to be able to assign the appropriate scene category to the given query image. For more



(a) Scene Recognition (b) Semantic Segmentation (c) Object Detection

Figure 2.4: Given a RGB image, visual scene understanding can involve: **(a)** scene recognition, **(b)** semantic segmentation, and **(c)** object detection [NKP18].

than a decade, scene recognition has been an active research area benefiting a wide range of applications, such as image retrieval [VS07], service robots [LT19], video surveillance [SD19], augmented reality [MXD+18], etc. Semantic segmentation is a computer vision technique that adds a category label to each pixel in an image, essentially splitting the image into regions containing meaningful items or fragments of objects. Semantic segmentation classifies each pixel to provide a more complete knowledge of the image [Tho16, GLGL18, WA19, MWY+22]. Object recognition is a method that identifies and categorizes individual objects within an image. It involves recognizing predefined object classes and determining which of these classes an object belongs to. Object detection takes a step further by detecting and localizing objects inside an image by providing bounding boxes around the detected objects. Object detection can identify and locate multiple objects of different classes within the same image [RP00, DLY+17, ZZXW19, ZCS+23]. In vision-based localization, the system can use different types of cameras to capture images for scene understanding, mainly:

- Mobile camera: In this approach, the subject that requires positioning carries a mobile camera that captures visual information of the surrounding environment. The camera's images are processed using advanced computer vision techniques to extract features and patterns that help establish the subject's location. This method is versatile as it can be applied to various entities such as individuals or robots.

- Static camera: This approach involves a network of static cameras positioned at predetermined locations throughout the indoor space. These cameras continuously monitor the environment and track the subject's movement by capturing visual data. While this method can offer high accuracy, it necessitates

careful camera placement and calibration to ensure comprehensive coverage of the space. It is particularly suitable for scenarios where an infrastructure is pre-installed.

Our primary emphasis will be on the task of scene recognition with mobile camera-based systems, where the user is equipped with a camera (*e.g.*, smartphone camera or other end-user devices built-in cameras) that can move during the localization process to capture images from the user's point of view. To summarize the advantages of vision-based localization, consider the following:

- Rich information: Cameras capture detailed visual information, including textures, colors, shapes, and structural elements of the environment. This rich data can be leveraged to create distinctive representations for accurate recognition and localization.

- Contextual information: Visual data provides contextual information about the surroundings, enabling the system to understand the scene's layout, structure, and the relationships between objects. This context aids in accurate localization.

- Low infrastructure requirements: Vision-based systems can operate with minimal infrastructure. Cameras are often already present in devices like smartphones, eliminating the need for extensive additional hardware.

- Compatibility with different environments: Vision-based techniques can be applied in a wide range of environments, making them versatile for various applications.

2.3.2 Fine-grained Indoor Scene Recognition for Localization

Indoor scenes are complex because of the diversity of objects and layouts, the problem of occlusion as well as the variability of lighting and viewing orientations. Therefore, achieving great indoor scene recognition is quite challenging. The goal is to develop algorithms and models that can effectively handle these obstacles, and therefore achieve more accurate and comprehensive scene interpretation in real-world indoor situations. Indoor scene recognition has received a lot of attention in recent years, because of the advances in robust computing systems, the rapid

growth of computer vision algorithms, and the emergence of minimalist interactive devices. In indoor scene recognition, the scope extends from recognizing the broader category of an indoor scene, such as a kitchen or office, to the finer task of identifying specific rooms within those categories. While recognizing scene types lays the foundation for understanding the overarching context, identifying specific rooms involves a more intricate analysis. This distinction becomes particularly important in real indoor environments, where multiple instances of the same scene type coexist but possess unique characteristics. This paradox highlights the necessity for systems that can understand the subtle variations that make each instance distinct.

Unlike outdoor space, a room-level location in which different rooms within a building are distinguished may be sufficient for most indoor location-based services. A room-level indoor localization system is mainly designed for people who need positioning solutions that are less susceptible to conditions that alter the waves in indoor areas, focusing on easy system installation and usage. This cost-effective system does not require any additional infrastructure to function; it may operate in indoor environments that do not have pre-installed beacons, transmitters, or receivers.

In a vision-based indoor localization scenario, a "known environment" refers to an indoor setting where the system has prior knowledge or information about the visual features of the environment. This information is usually obtained during a training phase, where the system is exposed to the environment, and images are collected and labeled with corresponding location information. The advantage of a known environment in vision-based indoor localization is that it allows for more accurate and efficient localization results. However, it is essential to note that maintaining an up-to-date and accurate dataset of the known environment is crucial for the system's success. Changes in the indoor environment can lead to inconsistencies between the known features and the actual scene, potentially impacting the accuracy of localization results.

Vision-based indoor localization, particularly indoor scene recognition, presents significant challenges. These include addressing the diversity of indoor scenes, the need for robust algorithms to handle variations in lighting, layout, and viewpoint, as well as the requirement of user-friendly systems. Additionally, due to the data-driven nature of vision-based localization solutions, significant computational resources are required. This thesis focuses on using DL techniques to address

the issues of vision-based indoor localization. Before providing a survey on DL for vision-based indoor localization, we give in the following an overview of available datasets for scene recognition.

2.3.3 Datasets for Scene Recognition and Their Limitations

The most widely recognized dataset is ImageNet [DDS+09] created to address the problem of object classification. This publicly available dataset includes 21,841 classes with more than 14 million images, providing the scientific community with a valuable data resource for computer vision tasks. Each image is annotated with an object class label following the WordNet hierarchy [Mil95]. The ImageNet dataset contains a set of manually annotated images with an approximate image size of 500×400 pixels. Since 2010, ImageNet has been at the core of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a famous benchmark for image classification and object recognition [KSH12].

Scene recognition stands as one of the pivotal tasks within the area of computer vision. The transition from object recognition to scene recognition represents a shift from the general recognition of object categories within images to the more challenging recognition and understanding of outdoor and indoor environments. For a scene classification problem, a dataset containing the corresponding images (*i.e.*, images of indoor or outdoor environments) is required. Outdoor scene recognition pertains to the recognition of scenes in open outdoor environments. Outdoor scene categories include a wide range of urban environments, such as bridges, skyscrapers, highways, campuses, and historic landmarks. Additionally, datasets for outdoor scene recognition include images from various natural landscapes, such as forests, lakes, caves, mountains, and parks. In contrast, indoor scene recognition focuses on recognizing scenes in enclosed spaces, such as rooms, offices, shops, classrooms, and homes. It often deals with scenes characterized by man-made structures and objects, such as furniture, appliances, and decor.

Researchers have been working on diversifying datasets to make classification algorithms more effective. An overview of publicly available datasets for scene recognition is represented in Table 2.1. The Scene15 dataset [LSP06] is a dataset comprising 4,448 gray-scale images categorized into 15 scene classes. These classes encompass 5 indoor scenes, such as offices, bedrooms, and kitchens, and 10 outdoor scenes like mountains, forests, and streets. Each class contains between 210

Table 2.1: An overview of popular datasets for scene recognition.

Dataset	Type of scenes	#Classes	#Images	#Images per class	Image size	Dataset size
Scene15 [LSP06]	indoor/outdoor	15	4,488	210 to 410	$\approx 300 \times 250$	81.88 MB
MIT Indoor-67 [QT09]	indoor	67	15,620	≥ 100	$\geq 200 \times 200$	2.4 GB
SUN397 [XHE+10]	indoor/outdoor	397	108,754	≥ 100	$\approx 500 \times 300$	37 GB
Places88 [ZLX+14]	indoor/outdoor	88	$\geq 88,000$	$\geq 1,000$	$\geq 200 \times 200$	–
Places205 [ZLX+14]	indoor/outdoor	205	2.5 million	5,000 to 15,000	$\geq 200 \times 200$	–
Places365-Standard [ZLK+17]	indoor/outdoor	365	1,803,460	3,068 to 5,000	$\geq 200 \times 200$	28.901 GB
Places365-Challenge [ZLK+17]	indoor/outdoor	365	8 million	$\geq 4,000$	$\geq 200 \times 200$	112.901 GB

and 410 images, and the average image size is 300×250 pixels. The MIT Indoor-67 dataset [QT09] covers indoor scenes including categories, such as stores, public places, and working places. This dataset contains 67 indoor categories with a total of 15,620 images, with approximately 100 images per class. The images within MIT Indoor-67 have a minimum image size of 200×200 pixels. The Scene UNderstanding 397 (SUN397) dataset [XHE+10] boasts a more extensive collection of 397 scene categories, encompassing indoor and outdoor scenes. Each category includes over 100 images, resulting in a dataset containing 108,754 images, with an image size of about 500×300 pixels. The Places dataset [ZLX+14, ZLK+17] stands as an extensive collection of scenes, including a rich spectrum of 434 scene categories. The images within the Places dataset have a minimum image size of 200×200 pixels. Places88, Places205, Places365-Standard, and Places365-Challenge are subsets of the Places dataset.

Using these public scene datasets for vision-based indoor localization has a number of significant constraints. These datasets may not fully represent the variety of lighting situations, viewpoints, and indoor layouts encountered in the real-world. Furthermore, the annotations in public datasets lack the granularity required for indoor localization systems. Public scene datasets, which include general indoor scene images, often have relatively broad labeling granularity. These datasets categorize images into high-level scene categories, such as office, kitchen, or bedroom. Datasets intended for localization demand a much finer granularity in labeling. Instead of high-level scene categories, they require detailed and room-level descriptions of the studied indoor environments. For example, they need labels like office A, office B, conference room, etc. Domain gaps can result from the difference between public datasets and the complexities of real indoor environments. These datasets are not compatible with the specific requirements of particular applications.

As indoor scene recognition is a key component of vision-based indoor local-

ization, creating real problem-specific datasets directly supports real-world applications. Solutions to address this challenge encompass diverse approaches. Real-world data collection entails engaging individuals to take videos or images of the studied indoor environments using smartphone camera or other end-user devices built-in cameras [ZDM+16, LZL+16, KKAMAA19]. This method guarantees the acquisition of a wide range of scenario representations. Real data can also be acquired through the deployment of a mobile robot, actively navigating through indoor environments [EKRS13, XCD19]. In synthetic image indoor localization, environments are generated through computer software. Simulated environments generated with 3D rendering software offer a powerful and controlled means to generate diverse and dynamic indoor scenes [LCZ+22].

2.4 Deep Learning for Vision-based Indoor Localization

Since applications that aid humans in understanding their surroundings are supported by indoor scene recognition systems, it is crucial to develop robust and trustworthy indoor scene recognition models. One useful source used for indoor localization is image analysis and classification [MMM+20]. In this section, we will start by a comprehensive overview on image classification using DL. We will then discuss the use of CNNs for indoor scene recognition. On this basis, we will dissect existing approaches and analyze the requirements and challenges that arise in the pursuit of seamless indoor scene recognition for localization purposes.

2.4.1 Deep Learning Fundamentals for Image Classification

A fundamental task in computer vision is image classification, which aims to classify images into predefined labels. DL has developed into a game-changer for tackling image classification issues over time, achieving astounding performance gains. CNNs, in particular, have demonstrated their capacity to learn hierarchical representations directly from raw pixel data, enabling them to perform exceptionally well on image recognition tasks. In what follows, we explore the fundamental ideas behind image classification, passing from traditional handcrafted features to learned features, and the essential elements that make them effective at image recognition tasks. Understanding these fundamentals is crucial for building effective and accu-

rate image classification systems in various real-world applications, such as vision-based indoor localization systems.

2.4.1.1 From Handcrafted Features to Learned Features

Supervised learning is a machine learning method in which a model is trained using labeled examples to classify or predict new unlabeled data. In the context of image classification, supervised classification involves using samples of known classes, known as training datasets, to teach a model how to recognize and classify unknown images or objects with unknown identities within an image.

Feature extraction is critical for distinguishing and differentiating images. Handcrafted features are manually extracted features from images to represent meaningful visual characteristics such as corners, edges, colors, or texture patterns. These features are engineered based on prior knowledge and understanding of the data and the problem at hand. Some notable local visual descriptors have been widely used in image classification, such as Scale Invariant Feature Transform (SIFT) [Lin12], Histogram of Oriented Gradients (HOG) [DT05], Speeded Up Robust Features (SURF) [BETVG08], and Bag-of-Visual-Words (BoVW) [YN10].

For image classification, visual characteristic regions are first detected in an image to get descriptors that are capable of distinguishing between the images. These handcrafted descriptors are then used as inputs to ML classification algorithms, such as SVM [LW18], KNN [Pet09], or Random Forest (RF) [Bre01], to discriminate between different classes. The performance of these ML methods is largely dependent on the quality and relevancy of handcrafted features that act as a bridge between the raw image data and the classifier.

The handcrafted features showed some success in image classification, but required expert domain knowledge for feature design. The handcrafted features have major limits. Handcrafted features extract a low-level representation of the data and hence cannot provide a conspicuous abstract representation, which is required for recognition tasks. Additionally, high-dimensional feature vectors can result from handcrafted features. Dealing with such high-dimensional data can be computationally intensive, requiring substantial memory and processing power. Furthermore, the handcrafted features capacity is limited and determined by a specified mapping from the data to the feature space, which is fixed independent of the needs of any identification challenge [TKT+22]. Effective handcrafted features necessitate domain expertise and a thorough understanding of the problem at hand. As a re-

sult, feature engineering can be a time-consuming and expert-dependent procedure. Due to the inherent limitations of traditional image classification methods, the exploration of novel approaches became imperative.

2.4.1.2 Convolutional Neural Networks for Image Classification

Over the last decade, handcrafted methods have been replaced by DL architectures, which often follow an end-to-end learning scheme in a supervised manner, as they are able to autonomously learn and extract valuable features directly from raw image data. The progress in DL techniques has reduced the need for manual feature engineering. In 1989, LeCun et al. used DL for recognizing handwritten digits and employed the back-propagation algorithm to train a neural network [LBD+89]. The rise of Graphics Processing Units (GPUs) to meet the demanding computational needs of DL algorithms, as well as improved DL architectures and the availability of large labeled datasets, are key factors contributing to DL's widespread adoption and improved performance [CPC16]. DL can outperform handcrafted feature extraction methods, improving state-of-the-art recognition results [LBH15].

CNNs are DL models that are specifically built for analyzing grid-like structured data like images. These models learn hierarchical representations of features from raw pixel data by using interconnected layers of neurons. This capacity enables CNNs to capture intricate visual patterns, outperforming traditional approaches in a variety of computer vision challenges like image classification tasks [ZYT17]. Various architectures have been proposed for general image classification, including AlexNet [KSH12], VGG [SZ14], GoogleNet [SLJ+15], Inception [SLJ+15], ResNet [HZRS16], DenseNet [HLVDMW17], and EfficientNet [TL19, XLHL20, PDXL21]. A typical CNN architecture is a hierarchical network composed mainly of convolutional layers, activation functions such as rectified linear units (ReLUs), pooling layers, and fully connected (FC) layers. The output of the last FC layer is usually fed into a softmax activation function for multi-class classification. The output of the softmax layer is a probability distribution vector over the different studied classes. The intermediate convolutional layers carry the important responsibility of feature extraction. Figure 2.5 presents an illustration of a typical CNN architecture.

Each CNN architecture has its unique characteristics and distinctive features. ResNet-32, for example, is a CNN architecture in the ResNet family [HZRS16]. This particular network is named ResNet-32 due to its 32-layer depth. One of the main

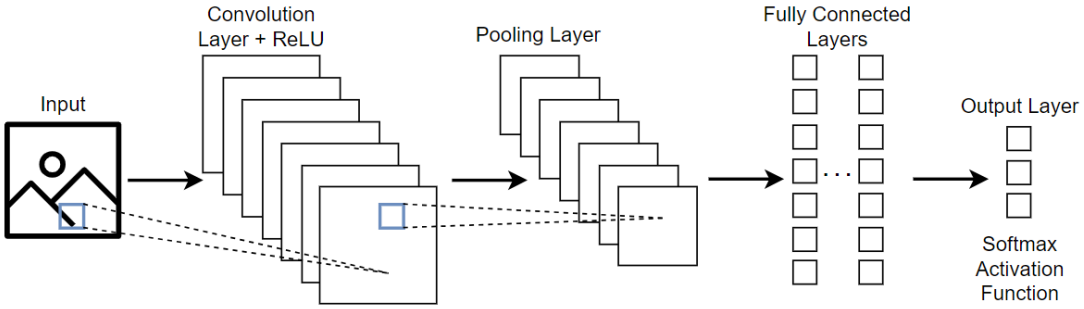


Figure 2.5: A typical CNN architecture.

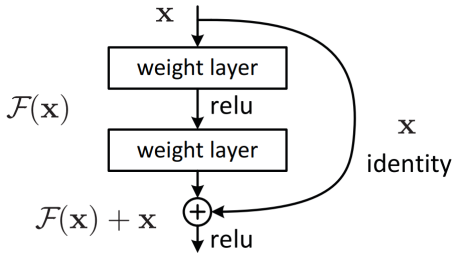


Figure 2.6: Residual learning: a building block [HZRS16].

innovations in ResNet-32, as in other ResNet models, is the implementation of identity shortcut connections as represented in Figure 2.6. These shortcuts allow information to bypass one or more intermediate layers, helping to mitigate the issue of vanishing gradient that typically affects the training of deep neural networks [GB10].

Before training CNNs, data preparation is an important step. The image pre-processing phase makes sure the data is of a quality and format that the CNN model can learn from. It is critical to resize images, as CNNs typically expect input images with a fixed size to maintain interoperability with FC layers. Pre-processing approaches such as resizing and normalization can improve model training by reducing computational needs and enhancing convergence [MMN22].

The dataset is typically divided into three subsets: training, validation, and testing datasets. The training dataset is used to train the CNN model. This dataset consists of labeled examples, where each example is an image along with its corresponding correct class label (*i.e.*, ground-truth) based on the image classification task at hand. Ground-truth is essential for quantifying how well the model's predictions are. The validation dataset is a separate dataset used to fine-tune the model's hyperparameters and to provide an estimate of the model's performance during training. The testing dataset contains a set of labeled examples like the validation dataset. These examples have not been used in training or validation. The testing

dataset is used to evaluate the final performance of the trained model and to get an estimate of how well the model generalizes to unseen data. Several public datasets have played a pivotal role in advancing the field of image classification, such as ImageNet [DDS+09], CIFAR-100 [KH+09], and Pascal VOC [EVGW+10]. These datasets have allowed researchers to train and evaluate models on large-scale diverse collections of images.

During training, input images from the training dataset are processed through the neural network. For each input image, the CNN predictive model generates a vector of scores (*i.e.*, softmax output probabilities), which is then compared to the corresponding ground-truth label. The difference between the predicted output and the true ground-truth label is quantified using a loss function (usually cross-entropy loss for classification tasks). The weights of the model are iteratively updated via a learning process until the output reaches the desired level of accuracy. The model's performance is evaluated using the validation dataset after each training epoch (*i.e.*, a full pass through the training dataset or a training batch). This evaluation involves feeding the validation data into the model and computing evaluation metrics (*e.g.*, accuracy and loss) to monitor the model's performance during training by assessing how well the model generalizes to data it has not seen during training. CNNs build a hierarchical representation of features from the input data, with the goal of minimizing a specific criterion presented as a differentiable cost function. This allows CNNs to learn both feature representation and feature encoding from images at the same time. During testing, the trained model is evaluated to assess its performance on new unseen test data. This phase determines how well the CNN generalizes its learned features to make predictions on real-world examples.

During inference, the capabilities acquired by training are put into practice. The trained CNN model uses the knowledge it gained through training to process each new input image (*i.e.*, a query image that the model has not encountered before) and produce an output (*i.e.*, a class prediction). The trained model is able to provide high-level feature representations for new input data related to the specific dataset and task on which it was trained [RW17]. The class with the highest probability in the softmax output is typically considered the predicted class by the trained CNN model for a given new input image.

CNNs, however, have limitations. For effective training, they require a large amount of labeled data, and the quality of the data and corresponding labels has

a significant impact on their performance. Unfortunately, many application fields do not have access to huge labeled data for training, leading to overfitting. Overfitting is a machine learning phenomenon in which a model gets highly specialized or fitted to the training data to the point where it performs badly on unseen or new data [SP22]. In other words, the model becomes too complex, capturing too much noise or random variation in the training data rather than learning the underlying patterns or generalizable features. These deep neural networks present additional challenges, such as the difficulty in ensuring properties like scale, rotation, or geometric invariance during training [XXZ+14, JSZ+15, TMU17]. Furthermore, CNN architectures with a large number of layers and parameters are frequently built for high accuracy on large-scale datasets. These deep architectures lead to significant computational and memory needs, making them difficult to deploy on resource-constrained embedded systems. To achieve optimal performance with CNNs, these challenges must be carefully addressed. This involves the strategic implementation of techniques, such as transfer learning and data augmentation, all while considering the adoption of lightweight CNN architectures when dealing with embedded systems as investigated in this PhD thesis.

2.4.1.3 Convolutional Neural Networks with Transfer Learning

With the evolution of DL, transfer learning has become a popular approach to solve new classification tasks with insufficient training datasets by fine-tuning pretrained CNNs [HBF19]. For example, a CNN model pretrained on large-scale datasets, such as ImageNet [DDS+09], can be fine-tuned with a training dataset containing images representing the target task. Typically, when pretraining CNNs with ImageNet, a subset of the dataset consisting of 1,000 categories and 1.2 million images is used [KSH12]. During transfer learning, the CNN weights are updated in an end-to-end manner in the training phase. Freezing the first layers refers to the process of fixing the weights and parameters of specific layers in a pretrained CNN while training on a new task. This process helps to preserve learned features from the source task that may be useful for the target task and reduces the number of trainable parameters in the network, which can significantly accelerate training and prevent overfitting. See Figure 2.7 for an illustration of CNN training on the target dataset using transfer learning.

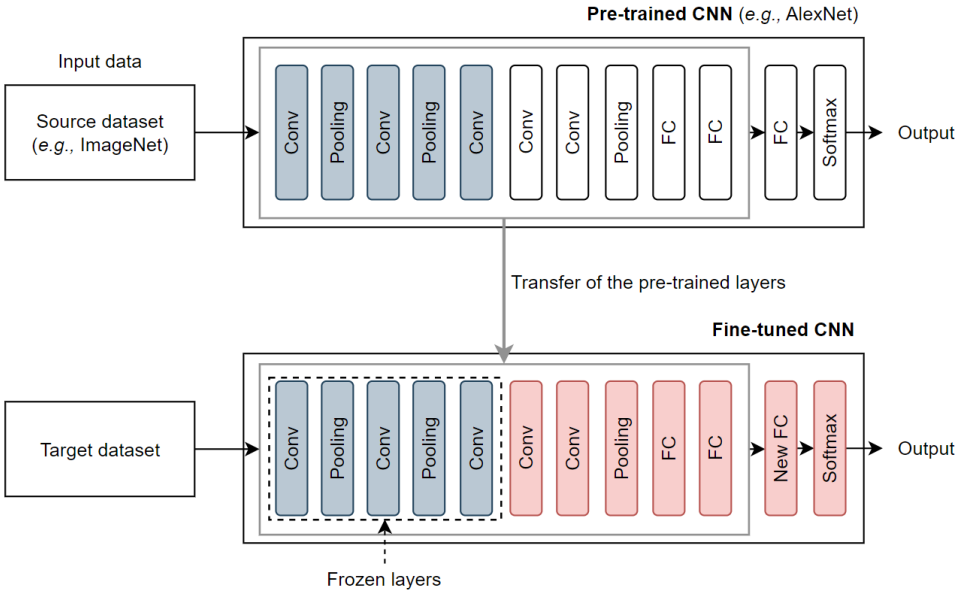


Figure 2.7: CNN training on a target dataset using transfer learning. A CNN is already pretrained on a source dataset. A target dataset is used to fine-tune the pretrained CNN to get a fine-tuned CNN for a specific task. Layers that are kept fixed during fine-tuning are not retrained to retain knowledge from the source dataset.

2.4.1.4 Convolutional Neural Networks with Data Augmentation

Data augmentation helps to represent a more comprehensive set of possible data features. There are several methods of data augmentation, including the application of geometric transformations, such as padding, scaling, rotation, flipping, etc. The majority of these image manipulation techniques are simple to implement. Data augmentation enables the CNN model to learn a wider range of image features and thus correctly predict the categories of unseen images while reducing overfitting [SK19]. Additionally, data augmentation is used to improve CNN invariance to image transformations, such as rotation and scaling [QRLB20]. CNNs are translation-equivariant, but not completely invariant to scale and rotation [XXZ+14, JSZ+15, TMU17]. The choice of the augmentation methods and the number of newly generated images is crucial since producing an excessive amount of augmented data might demand computational resources without proportional benefits.

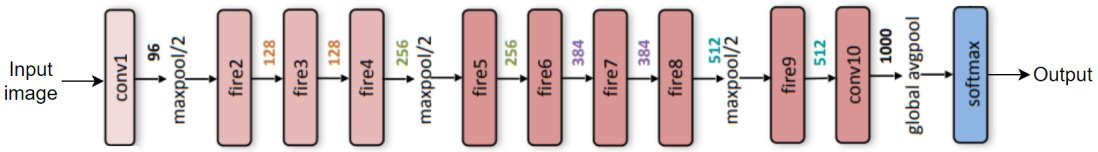


Figure 2.8: The architecture of SqueezeNet [IHM+16].

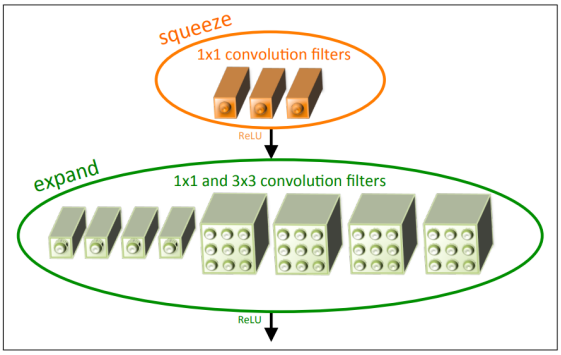


Figure 2.9: The organization of the convolution filters in the SqueezeNet fire module. In this example, the squeeze layer contains three 1×1 convolution filters and the expand layer contains four 1×1 and four 3×3 convolution filters [IHM+16].

2.4.1.5 Lightweight Convolutional Neural Networks

While it is known that the deeper the CNN model, the better the classification performance, lightweight (pretrained) CNN architectures are able to perform well with fewer layers and weights. Lightweight CNN models are intended for resource-constrained environments with low memory requirements for hardware circumstances and good performance for a variety of tasks, balancing between accuracy and efficiency. Examples of such lightweight CNN architectures are SqueezeNet [IHM+16], ShuffleNet [ZZLS18, MZZS18], MobileNet [HZC+17, SHZ+18, HSC+19], PeleeNet [WLL18], FBNet [WDZ+19], and GhostNet [HWT+20].

SqueezeNet [IHM+16] is a lightweight CNN architecture that uses 50× fewer parameters than AlexNet [KSH12] while achieving the same accuracy. SqueezeNet enhances accuracy while limiting the parameter count by adopting strategies such as replacing 3×3 convolution filters with 1×1 filters, reducing input channels, and downsampling late in the network. The fundamental component in SqueezeNet is the fire module. The fire module consists of a squeeze layer that employs multiple 1×1 convolution filters and an expand layer that combines 1×1 and 3×3 convolution filters. Both layers incorporate ReLU activation functions. See Figure 2.9 for the fire module representation. The SqueezeNet architecture contains 8 fire mod-

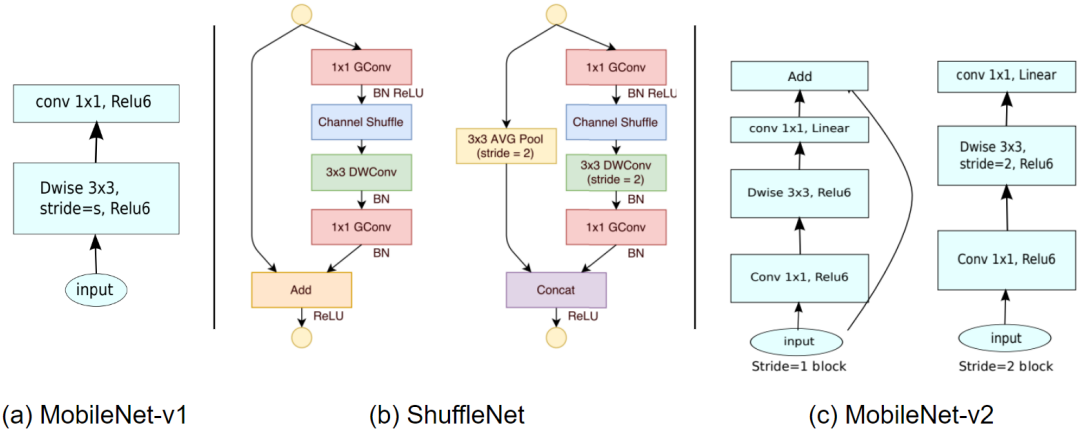


Figure 2.10: Comparison of convolutional blocks for different architectures. **(a)** MobileNet-v1 [HZC+17]. **(b)** ShuffleNet [ZZLS18]. **(c)** MobileNet-v2 [SHZ+18].

ules as illustrated in Figure 2.8. MobileNet-v1 [HZC+17] is another lightweight CNN that reduces the amount of parameters and operations by dividing the typical convolution into distinct depth-wise and 1×1 point-wise convolutions as represented in Figure 2.10a. In a typical convolution, filters operate across all input channels and combine them in a single operation. In contrast, depth-wise convolution employs distinct filters for each input channel and subsequently combines these using point-wise convolution. This decoupling of filtering and the feature combination lead to a reduced computational cost and a smaller model size. ShuffleNet [ZZLS18] is another small CNN architecture that uses an innovative technique known as channel shuffling to reduce the computational complexity and the memory footprint. This approach enables effective information exchange across different network channels. ShuffleNet uses group convolutions and point-wise group convolutions, as represented in Figure 2.10b, to find a balance between model size and accuracy. MobileNet-v2 [SHZ+18] is a version of the original MobileNet-v1 [HZC+17] built specifically for mobile and embedded vision applications. It uses inverted residual blocks and linear bottlenecks to improve both efficiency and performance. MobileNet-v2 employs, like MobileNet-v1, depth-wise separable convolutions as well as shortcut and residual connections to improve gradient flow and feature reuse during training. Figure 2.10c represents the convolutional blocks of MobileNet-v2. PeleeNet [WLL18] is an efficient architecture designed for deployment on resource-constrained embedded platforms. PeleeNet consists of a stem block to downsample the input image and four stages to extract features. PeleeNet has a compact architecture with a model size 66% smaller than MobileNet-v1 [HZC+17]. PeleeNet

outperforms the two versions of MobileNet [HZC+17, SHZ+18] in terms of accuracy and speed, performing 1.8× faster in ILSVRC 2012. FBNet [WDZ+19] is a family of compact and efficient CNN architectures discovered using the Differentiable Neural Architecture Search (DNAS) method. DNAS employs gradient-based techniques to optimize CNN architectures. The architecture of FBNet is inspired by MobileNet-v2 [SHZ+18], with a fundamental image model block that incorporates depth-wise convolutions and an inverted residual structure. GhostNet [HWT+20] is a lightweight CNN that is built around a Ghost module structure. The Ghost module is intended to extract additional features efficiently using lightweight operations. This method takes a foundational set of feature maps and uses cost-effective procedures to generate a number of ghost feature maps. Each ghost feature map contributes to a more thorough representation of the information embedded within the original intrinsic features.

2.4.2 Convolutional Neural Networks for Indoor Scene Recognition

Scene recognition stands as one of the pivotal tasks within the area of computer vision. This task encompasses both outdoor and indoor environments. Outdoor scene recognition pertains to the recognition of scenes in open outdoor environments. Outdoor scene categories include a wide range of urban environments, such as bridges, skyscrapers, highways, campuses, and historic landmarks. Additionally, datasets for outdoor scene recognition include images from various natural landscapes, such as forests, lakes, caves, mountains, and parks. In contrast, indoor scene recognition focuses on recognizing scenes in enclosed spaces, such as rooms, offices, shops, and homes. It often deals with scenes characterized by man-made structures and objects, such as furniture, appliances, and decor.

Indoor scene recognition began with traditional handcrafted feature extraction techniques, such as SIFT [LSP06, KPHB19, JQF+20] and HOG [LLLJ14], combined with typical ML algorithms. Researchers have investigated a variety of handcrafted features and classification models to classify indoor scenes based on color, texture, and shape information. The BoVW model also became popular for recognizing indoor scenes [YJHN07]. It entails displaying images as histograms of visual words derived from grouping and quantization of local image attributes. The BoVW model detects the distribution of visual words in an image and classifies them using clas-

sic ML algorithms like SVM [HDO+98]. Researchers have also included spatial relationships between objects or regions inside an image. Spatial pyramid matching techniques and Markov Random Fields were used to represent and use spatial context information for better recognition accuracy [MB14, XLL+18]. However, these approaches often failed to represent the hierarchical nature of visual information present in indoor scenes.

Thanks to the success of CNNs, starting with AlexNet in object classification with the large-scale ImageNet dataset [KSH12], research focus on scene classification has been diverted from handcrafted feature extraction methods to DL [ZLK+17, TWK17, ZLT+21]. CNNs have shown great promise for large-scale classification and detection tasks. CNN-based approaches predict the probability of scene categories directly from the entire scene image by extracting rich feature representations from these images. CNNs are designed to learn features from images in a supervised training procedure and can learn high-level feature representations of an image [KHB+16], as described in Section 2.4.1.

To improve the performance of scene recognition systems, researchers have been interested in replacing these traditional feature detection methods with deep neural networks, such as CNNs. Ongoing research in vision-based indoor localization focuses on improving scene recognition algorithms' accuracy, efficiency, and scalability by improving feature extraction techniques, developing robust matching algorithms, integrating DL approaches for visual representation and recognition, and exploring multi-modal approaches that combine visual information with other sensor modalities.

Several approaches have been proposed in literature for indoor scene recognition based on visual data. In [KHB+16], the authors proposed a feature representation based on rich mid-level convolutional features to categorize indoor scenes enabling more precise scene representation. In [HKBA16], the proposed approach combines a spatial pyramid encoding scheme with scale-invariant feature descriptors to capture both global and local information from indoor scenes. This approach enabled robust and invariant feature extraction, allowing for effective indoor scene classification. In [KHP17], the authors proposed implementing a spectral transformation as a convolution layer in the CNN model. Spectral features from images were used to capture unique signatures of different scene categories. In [SHC+19], the authors took into consideration the relationship between scenes and objects. The proposed scene recognition system uses object features extracted

with ImageNet CNN [DDS+09] and scene features extracted with pretrained Places CNN [ZLK+17]. In [SHK20], a lightweight architecture and effective feature extraction called FOSNet is proposed. FOSNet consists of PlaceNet for global context information extraction, ObjectNet for local features extraction and trainable fusion modules to enhance scene understanding. In [AASA20], the proposed approach applies fine-tuning on pretrained EfficientNet [TL19] versions to extract and learn discriminative features from indoor scene images.

The combination of multi-modal data, such as RGB images, depth maps, and semantic information, has become an active field of research to improve the robustness and discrimination capacity of indoor scene recognition. Combining several modalities allows for a more comprehensive knowledge of indoor scenes, which results in enhanced accuracy and scene understanding. In [WHA14], the authors introduced an innovative approach to indoor scene recognition that leverages RGB-D data. The proposed method uses object-level image representations and attributes to describe scene properties and learn scene-specific sub-dictionaries. In [ZWL16], the authors considered the relationships between RGB and depth data. The proposed pipeline involves the extraction of deep features using both an RGB CNN and a depth CNN, followed by a multi-modal learning layer. This approach takes into account inter-modality and intra-modality correlation for all samples, ensuring that the acquired features are both discriminative and compact. In [LCEVBGM20], a novel approach that takes advantage of both visual and semantic features is proposed, combining CNNs with semantic embeddings. In [YWLW20], the authors proposed T-Net, a T-like network that exploits both global and local features by multi-scale supervision. They include an image translation component and a pixel-level semantic segmentation annotations alongside the image-level labels. This joint supervision technique helps the model in identifying additional object regions in the images. In [MZM+21], a novel approach that uses object knowledge to improve indoor scene recognition is presented. The proposed method incorporates object-level features and object relations into the scene recognition system using an object feature aggregation module and an object attention module.

2.5 Requirements and Challenges for Indoor Scene Recognition

Indoor scene recognition poses specific requirements due the complexity of indoor environments and the diverse range of visual information present. Indoor scene recognition systems should be robust to variations in lighting conditions, viewpoints, and environmental changes. They need to handle different scenarios that can impact the appearance of the scene. As indoor environments can be large and complex, the system should scale well with the size of the environment and the number of possible scene categories. Moreover, real-time or near-real-time processing is essential for seamless user experiences in many applications. Indoor scene recognition systems should be capable of efficient and fast image analysis to provide timely responses.

The main requirement in indoor scene recognition is to understand and recognize input image data captured by a camera. As a result, robust features must be extracted from these images. DL approaches, particularly CNNs, have shown to have considerable potential in solving indoor scene recognition [SHK20, AASA20, CBLC20]. However, significant obstacles remain, mainly:

- Limited data availability: Because scene recognition is heavily data-dependent, it is critical to reuse knowledge learned from large-scale datasets using transfer learning. Learning solely from limited target dataset leads in low generalization. By applying transfer learning, CNNs pretrained on large-scale datasets (such as ImageNet [DDS+09, KSH12]) are fine-tuned with target scene datasets by making the last layers more task-specific [ZLT+21].

- Intra-class variability and inter-class similarity: Considerable variations between images of the same class and perplexing similarities between images of different classes make indoor scene recognition a difficult task. A high performance system should be able to deal with the inherent difficulty of indoor scenes [KHB+16].

- Different light conditions: Indoor lighting conditions can vary significantly, resulting in differences in illumination, shadows, and color tones. These lighting differences can have an impact on the appearance of objects, textures, and the overall qualities of a scene. Robust indoor scene recognition systems should be able to manage a variety of lighting situations.

- Moving furnishings and humans: Indoor scenes are dynamic, with moving furniture and present people, which alter the appearance and the layout of the scene over time.
- Occlusion of features: Objects or scene elements might be partially or entirely occluded, resulting in less informative features for indoor scene recognition. Occlusions may occur as a result of furniture, walls, or other objects' presence in the scene. Knowing that occluded features can provide essential information for scene recognition, dealing with this problem is critical.

It is critical to address these challenges in the context of indoor scene recognition in order to increase the accuracy and robustness of CNN-based models. Innovative solutions, such as data augmentation techniques [DPHB21], context modeling approaches [ZBK+17], and transfer learning strategies [AASA20], are constantly being investigated in order to overcome these limitations and fully exploit CNNs in indoor scene recognition.

2.6 Conclusion

In this chapter, we addressed indoor localization, providing an encompassing overview of diverse state of the art methods. Following that, we turned our attention to vision-based indoor localization. We delved into indoor scene recognition. We also investigated the use of DL for vision-based indoor localization, with a focus on CNNs, where we looked at existing approaches, public datasets, and the limitations and challenges that result. Additionally, we provided a comprehensive overview of the key steps involved in building an indoor localization system.

Chapter 3

Smartphones in Indoor Localization Systems

Sommaire

3.1 Introduction

The integration of smartphones into indoor localization systems marked the beginning of an era of innovation in localization technology. Due to their widespread use

and the variety of sensors they carry, smartphones are uniquely positioned to play a key role in indoor localization systems. By exploring the symbiotic relationship between smartphones and indoor localization, this chapter discusses the capabilities of smartphones sensors, as well as their limitations, when used for location estimation. This chapter also includes some examples of existing smartphone-based indoor localization systems. We also present the DL frameworks interoperability for smartphone deployment as well as the advantages and disadvantages of the current computing approaches.

3.2 Smartphone-based Indoor Localization

The rise of smartphone-based indoor localization represents a significant advancement in the pursuit of location-aware services within indoor environments, ranging from malls, hospitals, airports, and beyond. Smartphone-based indoor localization has several compelling benefits that make it a promising and practical solution for a wide range of applications. Smartphones are widely owned and carried by individuals, making them an accessible and low cost platform for indoor localization. Using devices already in people's possession simplifies the implementation and adoption of localization solutions. On the other hand, the majority of indoor localization systems need the use of external sensors. While some systems just require a single additional sensor, others require multiple sensors. In this context, a smartphone-based indoor localization system is exceptionally user-friendly. Compared to deploying dedicated infrastructure, such as WiFi access points, Bluetooth beacons, and RFID tags, utilizing smartphones as localization tools minimizes additional costs. This is particularly advantageous for large-scale implementations as smartphone-based localization systems can easily scale to cover large areas. In this section, we investigate the use of smartphone sensors for indoor localization and present some existing systems.

3.2.1 Evolution of Mobile Phones and Smartphones

The evolution of mobile phones and the sensors embedded within them has been nothing short of remarkable. The number of integrated sensors in mobile phones has increased dramatically over the past years as illustrated in Figure 3.1. The year 1992 witnessed the advent of mass-produced Global System for Mobile Commu-

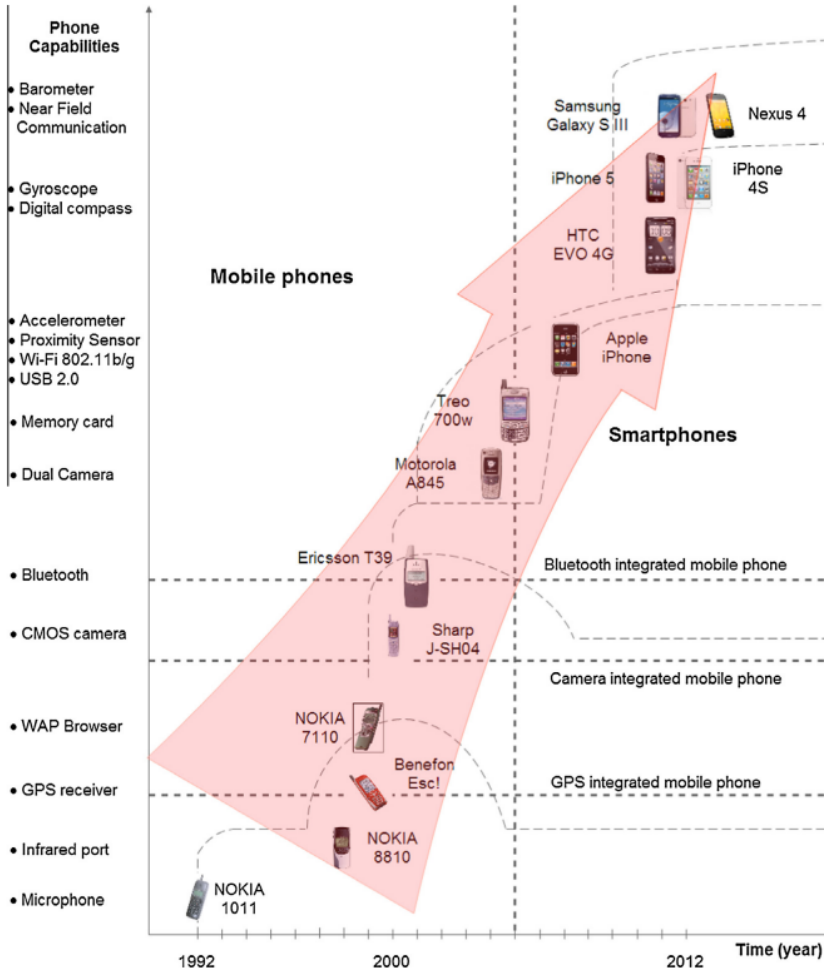


Figure 3.1: The evolution of mobile phones and smartphones from 1992 to 2012 [DDVPR13].

nication (GSM) phones with the introduction of the Nokia 1011. By 2000, mobile phones had become more sophisticated, with many models featuring integrated infrared ports, GPS receivers, and cameras. In 2002, the Ericsson T39 integrated Bluetooth. In 2004, the Motorola A845 was the first mobile phone to integrate a dual camera. In 2007, the Apple iPhone was released. The iPhone was one of the first smartphones to feature a touchscreen display and a variety of integrated sensors, including an accelerometer, a proximity sensor, and a wireless sensor. By 2012, smartphones had become ubiquitous, and many models featured a wide range of integrated sensors, including gyroscopes, digital compasses (magnetometers), barometers, and Near Field Communication (NFC) chips [DDVPR13]. The evolution of smartphone sensors continued to accelerate in the past decade, with the introduction of even more sophisticated and specialized sensors. Fingerprint sensors gained

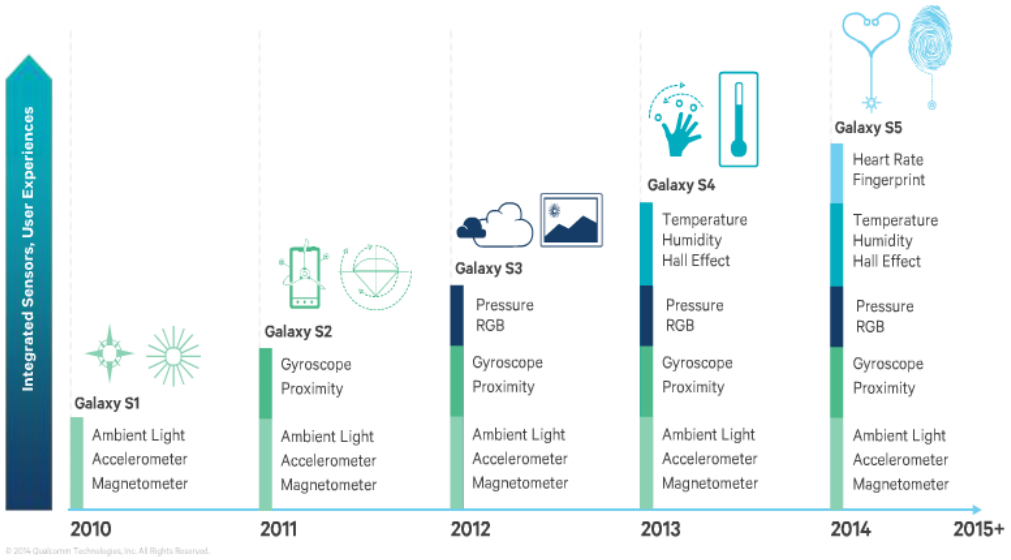


Figure 3.2: Samsung Galaxy smartphones sensor growth.

widespread adoption with the iPhone 5S [1] , elevating security measures and unlocking mechanisms. In 2014, the Samsung Galaxy S5 integrated heart rate sensors as represented in Figure 3.2 from Qualcomm [2]. In more recent developments, the iPhone 11 integrated a UWB sensor [3], while the iPhone 12 Pro incorporated a Light Detection and Ranging (LiDAR) sensor [4]. These additions have significantly broadened the spectrum of capabilities in these devices [CVPA22, DPD23, HKPH23]. Today, smartphones serve as versatile hubs of sensing, processing, and connectivity. This combination has unlocked a world of novel applications that were previously unattainable with standalone devices. As we move forward, smartphones continue to evolve, and so do sensor technologies.

3.2.2 Smartphone Sensors for Indoor Localization

The growth of smartphone sensors has been remarkable, converting mobile devices from simple communication tools to versatile, context-aware computing platforms.

[1] Apple Touch ID technology: `https://support.apple.com/en-us/105095`, accessed on 22 November 2023.

[2] Qualcomm; Website: `https://www.qualcomm.com/news/onq/2014/04/behind-sixth-sense-smartphones-snapdragon-processor-sensor-engine`, accessed on 30 October 2023.

[3] Apple UWB sensor availability: `https://support.apple.com/en-us/HT212274`, accessed on 22 November 2023.

[4] Apple LiDAR camera availability: `https://developer.apple.com/documentation/avfoundation/additional_data_capture/capturing_depth_using_the_lidar_camera`, accessed on 22 November 2023.

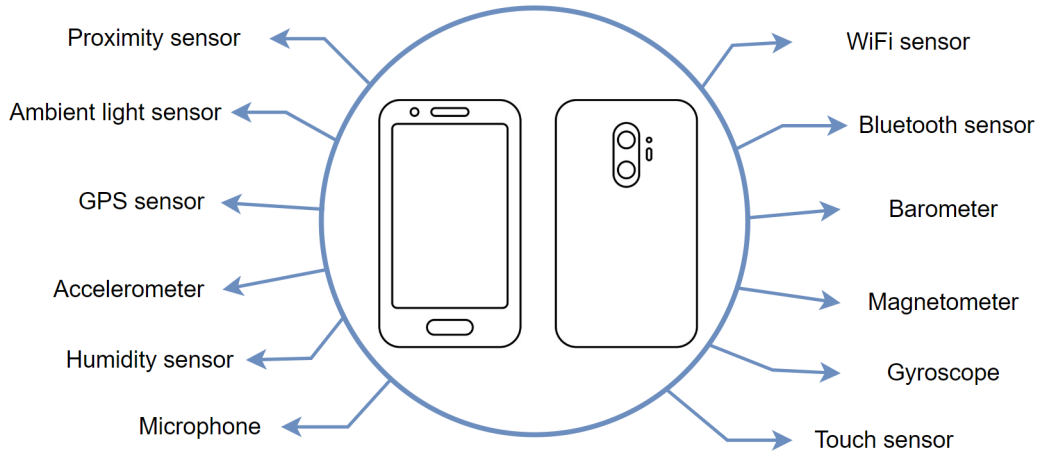


Figure 3.3: Some built-in smartphone sensors [LTP17].

This evolution has been driven by technological breakthroughs and an increasing demand for different applications. Smartphones are equipped with many sensors that can be used and optimized for accurate indoor location estimation. These sensors, originally integrated to enhance the user experience and enable various functionalities, have found new applications in indoor localization in real-world environments [AHP20]. Some of the key sensors, represented in Figure 3.3, include:

- Accelerometer: This sensor measures acceleration across all three axes, aligned with the smartphone's body frame caused by movement (dynamic force) or by gravity (static force). When at rest, an accelerometer measures an acceleration equal to the standard gravitational acceleration on the surface of the Earth. It is used in axis-based motion sensing. In indoor localization, it is crucial for tracking movement and changes in position.

- Gyroscope: This sensor measures angular velocity, indicating the direction and speed that the smartphone is spinning about each of the three axes. It provides information about the orientation and angular changes of a device, which can contribute to more accurate localization.

- Magnetometer: This sensor detects the local magnetic field strength and direction along the three axes of the smartphone, enabling the determination of direction and orientation relative to magnetic North. It plays a vital role in magnetic fingerprints localization as well as orientation estimation and can

assist in positioning calculations with other inertial sensors (*i.e.*, accelerometer and gyroscope).

- Barometer: This sensor measures atmospheric pressure, which correlates with changes in altitude. By combining barometric data with other sensor inputs, it is possible to estimate changes in floor levels within a building.
- WiFi and Bluetooth: These sensors help in WiFi and Bluetooth signal strength measurements. By analyzing signal strengths from nearby WiFi access points or Bluetooth beacons, smartphones can estimate their proximity to known reference points in indoor environments.
- Camera: This visual sensor is of particular significance for indoor localization. Advanced image processing and computer vision techniques can be employed to extract visual features from captured images, aiding in accurate localization.
- Microphone: This sensor captures and records audio. It can detect sound waves in the environment and convert them into electrical signals, which are then processed by the smartphone's audio system for various applications. This sensor can be used with other sensors like WiFi, Bluetooth, or inertial sensors to enhance localization accuracy.

Two major smartphone-based localization approaches can be identified. The first approach focuses on directly using the various built-in smartphone sensors like camera, microphone, accelerometer, gyroscope, and magnetometer. The second approach involves exchanging data and measurements between the smartphone and built-in wireless or wired communication technologies. In the following, we explore some smartphone-based indoor localization systems.

3.2.3 Camera-based Indoor Localization

The ability of smartphones to share the captured images opens up the possibility of using the camera for localization. In fact, using smartphone camera for indoor localization is a very interesting and viable strategy [SHC+11]. Camera-based indoor localization leverages the embedded cameras in smartphones and their good quality images to determine the user's indoor location by analyzing visual data of the environment. This category includes techniques such as indoor scene recognition and

object detection [WKM11, MMFW14, ZLT14, MNP+16], SLAM [LSY+22], QR-Codes scanning [CMGCCL+11, PMMRSP13], Visible Light Positioning (VLP) [RF21], etc. These approaches are mainly based on computer vision techniques and algorithms to identify unique visual features within indoor spaces [JD17, GWCZ18, AHP19]. The solution to use depends on several factors like the complexity of the indoor environment, the level of precision and accuracy required, the available infrastructure as well as the accessible devices and data. We describe in the following some existing camera-based localization systems that use conventional approaches or DL solutions for indoor scene analysis and understanding (see Sections 2.3 and 2.4 for detail on vision-based indoor localization and DL).

3.2.3.1 Examples of Existing Systems

DeepMoVIPS (Deep Mobile Visual Indoor Positioning System) [WHS16] is a room-level indoor positioning system that leverages the image classification capabilities of deep CNNs. DeepMoVIPS employs AlexNet [KSH12], pretrained on ImageNet dataset, and extract feature values from its FC layer. Various classifiers (like Random Forests [Bre01] and Naive Bayes [R+01]) are trained on transformed images using these features. The evaluation encompasses rooms in an office building environment, demonstrating the system's effectiveness in real-world indoor localization smartphone application.

VizMap [GGL+16] is an indoor localization system for visually impaired people. It employs computer vision with crowd-sourcing to gather indoor visual POIs such as posters, signs, and exit doors. VizMap initiates the process by utilizing videos captured by on-site volunteers using smartphones (or wearable devices), which are then used to construct a 3D spatial model. Remote crowd workers semantically label the collected video frames. These labels are integrated into the 3D model, creating a spatial representation of the environment that can be queried. Users can be localized by capturing images using their smartphones. VizMap is based on SIFT features for image matching and achieves indoor localization with sub-meter accuracy, enabling users to interact with their surroundings effectively. See Figure 3.4 for an overview of the VizMap system.

DeepSpace [ZDM+16] is an approach based on deep CNNs for indoor positioning within MIT (Massachusetts Institute of Technology) campus buildings adapted to smartphone camera use. DeepSpace utilizes two spatial-scale CNNs models; one for building-level recognition (low spatial resolution) and another for room-level

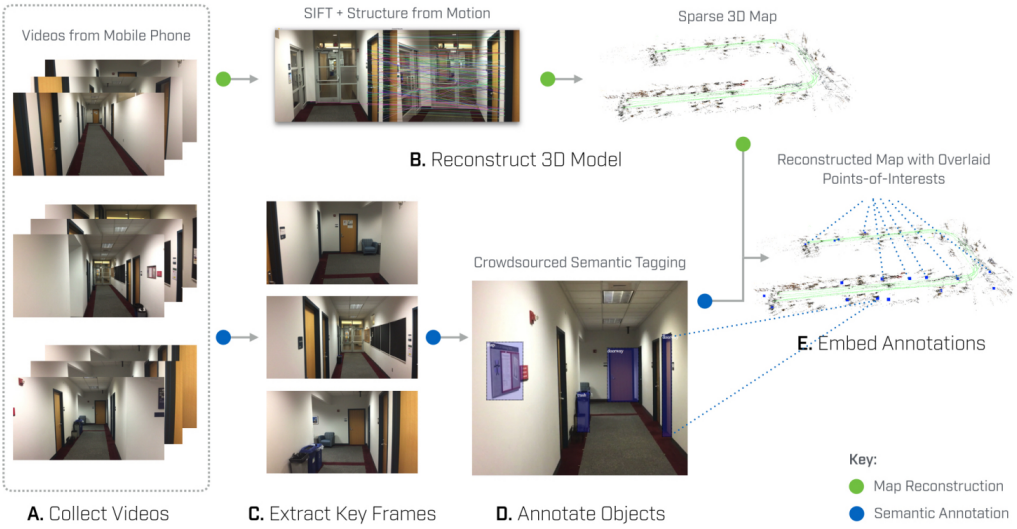


Figure 3.4: The VizMap system [GGL+16]. VizMap collects videos from sighted volunteers **(A)** and constructs a sparse 3D model of the environment **(B)**. At the same time, clear key frames are extracted **(C)** for the crowd to annotate points of interest **(D)**. Finally, the crowd labels are embedded into the generated points cloud **(E)**, shown here as blue squares.

recognition (meter-level area precision). The CNN model design draws inspiration from architectures like AlexNet [KSH12] and NIN [LCY13]. The research involved collecting video data within MIT campus buildings using a GoPro camera, from which images were extracted for the training phase. In the inference phase, the building-level CNN initially predicts the input image's building category and then directs it to the corresponding room-level CNN model for high-precision localization. See Figure 3.5 for an overview of the DeepSpace system.

The study in [XCL+18] describes an indoor positioning system that can accurately find the user's location in a large indoor environment. This system leverages ordinary static objects (*e.g.*, doors and windows) as reference points to determine positions. The method first uses a smartphone's camera to detect these static objects and then calculates the smartphone's position based on DL and computer vision algorithms. The proposed system relies on Faster Region-based CNN (Faster R-CNN) [RHGS15] end-to-end network for image recognition and feature extraction to identify static objects within an indoor environment. See Figure 3.6 for a diagram of the indoor positioning approach. Experimental validation conducted in an art museum environment achieved results with a 1-meter range precision.

The paper [LCL+20] outlines a comprehensive method for precise indoor visual positioning using smartphones cameras. First, a sequence of RGB images

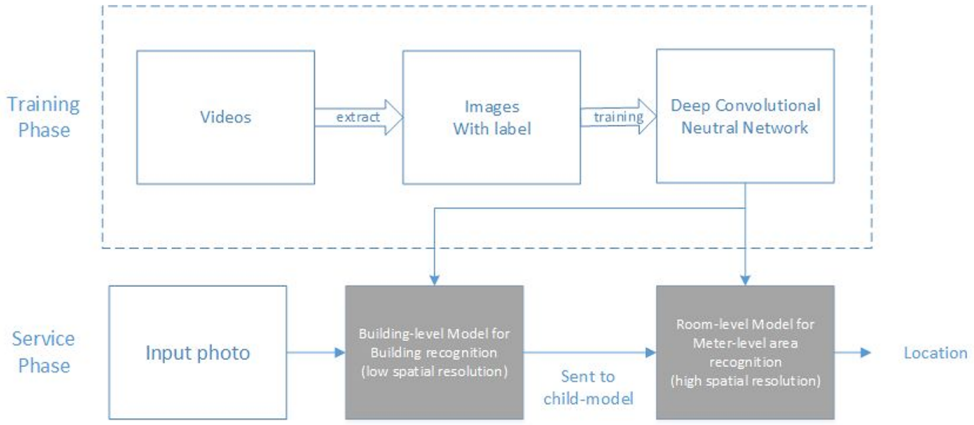


Figure 3.5: DeepSpace system diagram [ZDM+16].

are collected from the indoor environment using a Sony ILCE-5000 camera, from which an indoor precise-positioning-feature database is generated. The system employs a traditional SURF [BETVG08] point matching strategy along with multi-image spatial forward intersection techniques. Subsequently, the relationships between SURF [BETVG08] feature points in smartphone positioning images and 3D object points are established through an efficient similarity feature description retrieval method. This method employs a novel matching error elimination technology based on Hough transform voting [DH72] to obtain a reliable set of matching point pairs. Finally, the intrinsic and extrinsic parameters of the positioning image are calculated using Efficient Perspective-n-Point (EPnP) [LMNF09] and Bundle Adjustment (BA) [TMHF00] methods, yielding to the smartphone's precise location. Basically, users capture an image using their smartphone's camera, feature extraction is then performed on the smartphone, and the image features and camera information are transmitted to the server. The server utilizes the pre-established positioning feature database to calculate the accurate pose of the captured image. Finally, the precise pose information is transmitted back to the user's smartphone and displayed, enabling real-time self-positioning of the smartphone camera. See Figure 3.7 for an overview of the indoor positioning system. This method offers a robust and accurate solution for indoor visual positioning using standard smartphones cameras and a well-structured image feature database.

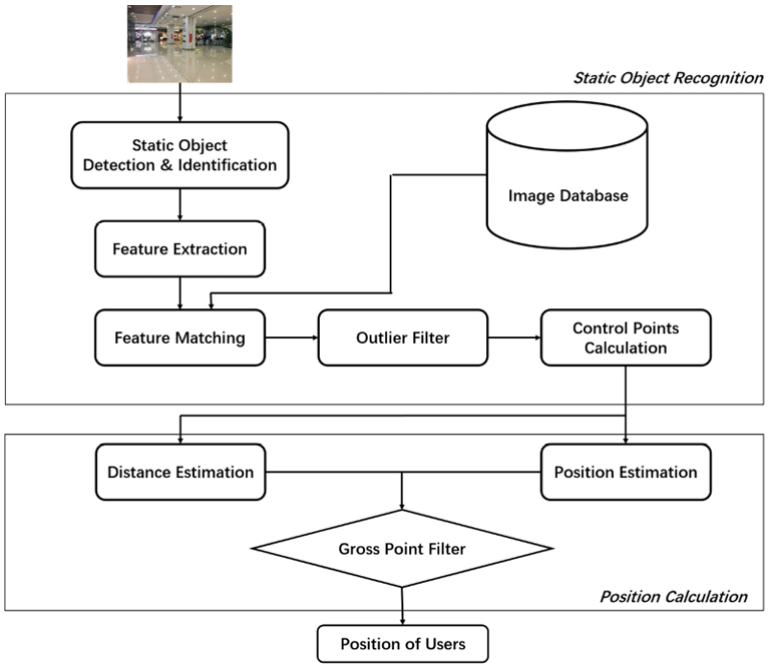


Figure 3.6: Diagram of the indoor positioning approach proposed in [XCL+18].

3.2.3.2 Challenging Problems

Although not expensive, smartphone camera-based localization has several associated challenges that can undermine the system's accuracy. Vision-based indoor localization performs poorly in many situations, such as corridors and similar rooms, where features are frequently indistinguishable and repetitive, making it difficult to find the correct location (as described in Section 2.5). To achieve high accuracy and to compensate for the error, one option is sensor fusion techniques. Another downside of the vision-based system is the storage capacity required for creating the database of 3D maps, labeled images or features. Additionally, significant computational resources are required throughout the localization phase to perform image feature extraction, matching and classification [AHP20] (see Section 3.3.2).

3.2.4 Magnetic-based Indoor Localization

The magnetic field of the Earth is a natural phenomenon. It maintains a consistent magnetic field strength across vast distances and does not undergo sudden changes over short shifts like a few meters. However, the presence of ferromagnetic materials within indoor environments introduces anomalies [ZWWN15, SGD13]. These

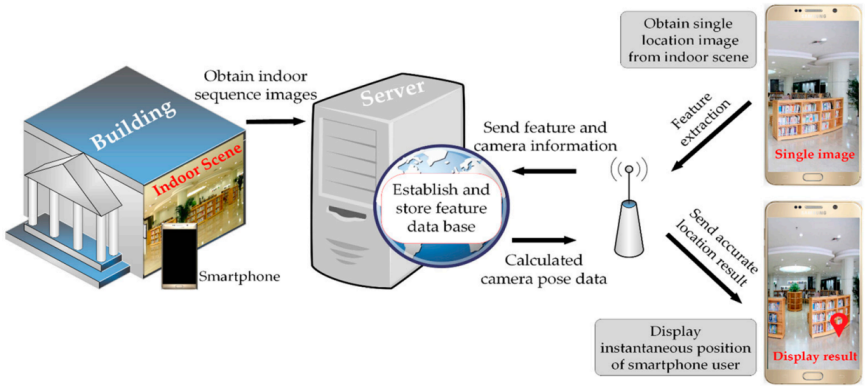


Figure 3.7: The workflow chart of the indoor visual positioning system proposed in [LCL+20].

anomalies, detectable by a magnetometer, can serve as fingerprints, enabling location estimation. The magnetic field has been used for localization and tracking in a variety of applications [AHP20, TAC+21, OAM22].

3.2.4.1 Examples of Existing Systems

Integration of magnetic field data as fingerprints, along with visual information for indoor localization, was studied in [LZL+16, AHP19]. Both papers present solutions based on computer vision and DL approaches.

In [LZL+16], the authors introduced an indoor localization and tracking approach called VMag that leverages the fusion of magnetic and visual sensing to enhance positioning accuracy. VMag employs a context-aware particle filtering framework for user tracking and a neural network-based method to extract deep features from input data. In the training phase, image and magnetic signals fingerprints for each location are collected manually using smartphones (in vertical mode and pointing into the main direction of the path). These measurements, as well as the floor plan of the building, are then stored at the server end. The positioning deep features extraction models are built and trained on the server side. In the inference phase, a user holds his smartphone vertically. After each step, the application takes an image and measures the magnetic field at its current location. These measurements are sent to the server for localization based on the trained models. The motion data and the trajectory data are also sent to the server to support the tracking of the user. Extensive experiments demonstrate VMag's effective performance in four diverse indoor environments, including a laboratory, a garage, a canteen, and an

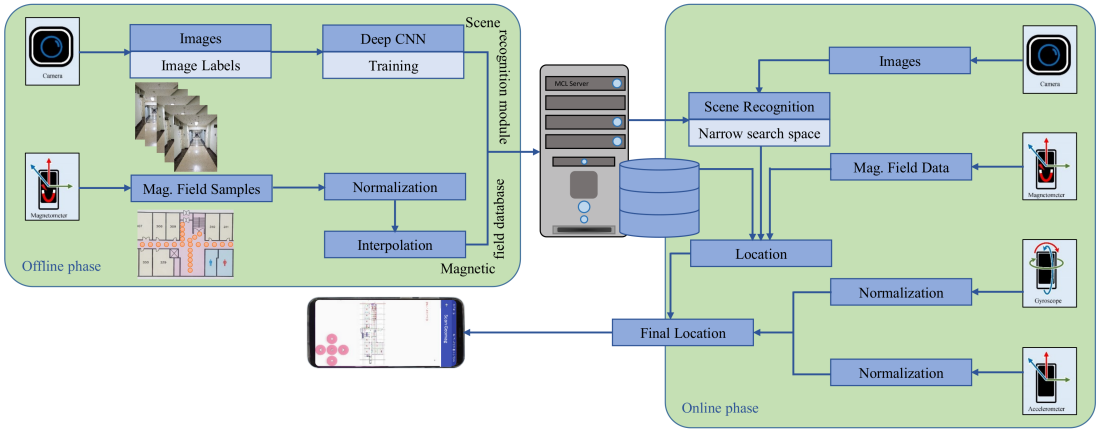


Figure 3.8: Architecture of the indoor localization approach proposed in [AHP19].

office building.

In [AHP19], an indoor localization approach is presented based on multi-sensor fusion. First, the indoor scene is recognized using a CNN, decreasing localization errors and restricting the search space by identifying the unique floor. Then, the identified scene is employed to narrow down the search space in the magnetic field database to lower the localization error. A modified K-NN algorithm computes the user's current location, which is adjusted using PDR data, and an expanded Kalman filter enhances accuracy even further. Experimental results show that the approach succeeds regardless of the smartphone used for localization. See Figure 3.8 for the architecture of the proposed approach.

3.2.4.2 Challenging Problems

Smartphone sensors are an important component in the mobile computing domain, serving as a platform for new applications. However, the accuracy of these sensors is critical for such applications. A smartphone magnetometer is a Hall effect sensor that perceives magnetic fields in an active manner [Chi13]. Artificial and natural magnetic fields are numerous and variable. Thus, the magnetometer sensor (digital compass sensor) of smartphones must be recalibrated regularly to reanalyze the present magnetic fields and determine where the North is. Magnetic field measurements are primarily dependent on:

- The device: Sensors' measurements are imperfect and imprecise. Different sensors have varying precision, sensitivity, and stability. Various built-in sensors and algorithms utilized by smartphone manufacturers lead to different

magnetic field measurements.

- The user's surroundings: Data uncertainty is caused not only by noisy measurements but also by the surrounding environment. Other electronic devices commonly found inside buildings cause interference and magnetic perturbation. The omnipresent magnetic field is disrupted by ferromagnetic materials used in buildings, affecting magnetic field measurements and causing inaccurate direction and position information.

All the above challenges may affect the performance of localization systems relying on magnetic field data. Thus, proper stability management is critical for achieving accurate localization results that are closer to situations in reality.

3.2.5 WiFi-based Indoor Localization

The widespread availability of Wireless Local Area Network (WLAN) infrastructures has accelerated the use of WiFi-based systems for indoor localization. The WiFi technology offers a cost-effective solution for localizing a diverse range of WiFi-compatible devices (*e.g.*, smartphones and tablets) without necessitating additional software installations [XZYN16, AMCHC20]. In the following, we describe some existing WiFi-based localization systems and delve into the inherent challenges associated with these systems.

3.2.5.1 Examples of Existing Systems

SWiN (Self-evolving WiFi-based Indoor Navigation) system is a real-time lightweight indoor navigation solution for smartphone implementation [ZHSS19]. SWiN retrieves both the static and dynamic WiFi signal features such as the scanned access point list, the variations of signal strength, and the access point's relative strength order. SWiN uses the leader-follower structure. The authors presented a novel step-constrained hybrid synchronization algorithm that allows navigation based on user motion patterns. Furthermore, an updating mechanism ensures the system's long-term utility. See Figure 3.9 for a diagram of the SWiN system. Experimental results obtained in a five-story office building and a two-story shopping mall affirm the effectiveness of the SWiN system deployed on smartphones.

In [GCY+19], the authors proposed an indoor smartphone localization and tracking solution based on WiFi technology. The system uses a hybrid algorithm

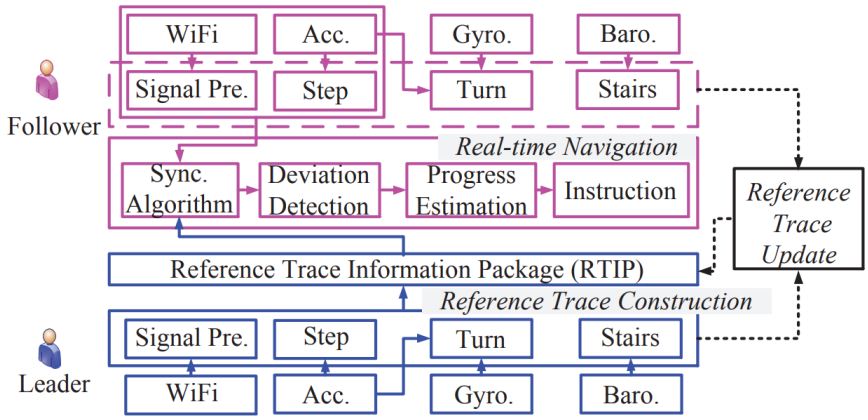


Figure 3.9: SWiN system diagram [ZHSS19].

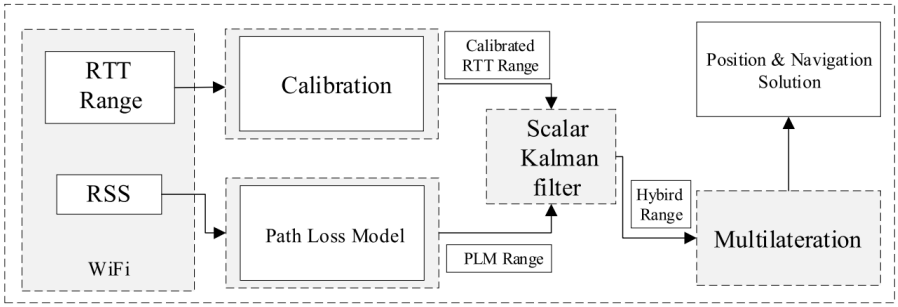


Figure 3.10: Architecture of the indoor localization system proposed in [GCY+19].

based on the WiFi Round Trip Time (RTT) and RSS to improve the positioning accuracy and scalability. A multilateration method is used to determine the position of the smartphone. Multilateration is a localization method that converts signal ranges to a position [LDBL07]. See Figure 3.10 for the proposed localization system architecture. Experiment results show that the proposed system outperformed the classic fingerprinting approach in terms of accuracy and update time.

3.2.5.2 Challenging Problems

WiFi-based indoor localization systems confront several challenges that can reduce their applicability and accuracy. The ubiquitous availability of WLAN infrastructures in indoor environments has led to the widespread use of WiFi technology for indoor localization. However, when such infrastructures are not available, the deployment of transmitters/receivers is required. The installation and the maintenance of these infrastructures incur significant costs. Additionally, WiFi

signals may fluctuate and vary due to factors such as temperature changes and moving objects. Furthermore, signal attenuation induced by static components, such as walls, doors, and furniture, presents a significant challenge for this technique [ACH18, ACHC20].

3.2.6 Bluetooth-based Indoor Localization

Bluetooth-based indoor localization relies on the Bluetooth technology to estimate a user's position within an indoor environment. The Bluetooth technology is widely available in low-cost devices, such as smartphones and tablets, making it a popular choice for indoor localization. In the following, we describe a range of existing Bluetooth-based localization systems and delve into the challenges that these systems face.

3.2.6.1 Examples of Existing Systems

In [SP20], the authors investigated an indoor positioning system for museums that relies on Bluetooth Low Energy (BLE) beacons. The interactive smart museum system uses an RSS-based technique to estimate the location of the visitor in the museum. The RSS from the BLE beacon is used to calculate the distance between the beacon and the receiver (*e.g.*, a smartphone). Trilateration is employed to estimate the receiver's location within the indoor environment. An Android application was developed to test the proposed system. To improve the localization accuracy, the application uses a simple Kalman filter, which is computed directly on the smartphone. Experimental results show the effectiveness of the developed solution.

In [PACK22], the authors introduced a novel approach for indoor smartphone localization based on the joint use of BLE beacons, RSSI values, fingerprinting, and neural networks. The neural network architecture utilized in this work follows the framework outlined in [KSV16], with certain modifications identified through rigorous validation processes. The neural network assists the system's response to changes in RSSI values. See Figure 3.11 for the architecture of the indoor localization system. Results show the significance of the proposed localization system.

3.2.6.2 Challenging Problems

A notable drawback of Bluetooth-based indoor localization lies in its execution of the device discovery procedure during each location estimation, leading to a sub-

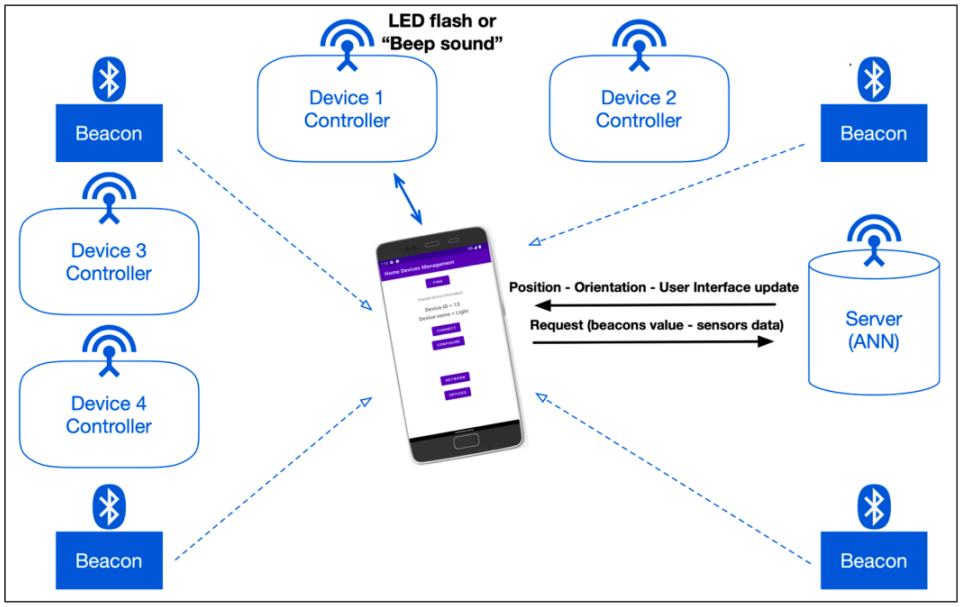


Figure 3.11: Architecture of the indoor localization system proposed in [PACK22].

stantial increase in localization latency and power consumption. This latency is particularly unsuitable for applications requiring real-time localization. Moreover, Bluetooth's short-range necessitates a dense deployment of Bluetooth beacons for achieving accurate positioning, leading to a higher cost for the required infrastructure [ZLXQ18].

3.3 Deep Learning for Smartphone-based Systems

Smartphones are widely used and easily accessible devices. With regard to processing power, memory size, and battery life, these devices still have limited resources. This section delves into the complexities of deploying DL models on smartphones, exploring the challenges posed by computational constraints and unveiling innovative solutions to overcome them.

3.3.1 Deep Learning Frameworks Interoperability

DL frameworks serve the dual purpose of developing DL models and deploying these models. DL model development refers to the process of designing, training and testing a DL model to perform a specific task. On the other hand, DL model deployment entails making a trained DL model accessible for use in real-world ap-

Figure 3.12: Deep Learning Frameworks interoperability.

plications (*i.e.*, inference). It is the transition from the development phase to the practical integration of the model into systems, devices, or applications that can make predictions based on new unseen data. Frameworks offer different levels of abstraction and flexibility for DL. The appropriate framework should align with the requirements of the task at hand, providing the required tools and functions for efficient model development and deployment. Several DL frameworks have been developed by leading technology companies for the deployment on smartphones and other resource-constrained platforms. Among these mobile DL frameworks are TensorFlow Lite by Google, PyTorch Mobile by Facebook, and Core ML by Apple [XLL+19]. Our focus revolves around exploring the synergies between MATLAB as a development framework and TensorFlow Lite [5] as a deployment framework, with a particular emphasis on achieving interoperability through the Open Neural Network Exchange (ONNX) [6] format.

Building DL models using MATLAB, provided by MathWorks, is made accessible through specialized Toolboxes [7], which encompass an extensive set of functions and tools designed specifically for DL. MATLAB provides an environment for data preparation as well as designing, training, and testing deep neural networks. MATLAB supports interoperability with other open-source DL frameworks such as ONNX. ONNX, jointly developed by Facebook and Microsoft, is an open format for representing DL models that allows for interoperability among various DL frameworks. ONNX allows models to be easily exported from one framework, such as MATLAB, and imported into another framework, such as TensorFlow Lite. TensorFlow Lite, introduced in late 2017, is a lightweight version of the TensorFlow [8] framework designed for mobile and embedded devices. TensorFlow Lite is specifically tailored

[5] TensorFlow Lite: `https://www.tensorflow.org/lite`, accessed on 23 September 2021.
[6] Open Neural Network Exchange: `https://onnx.ai/`, accessed on 26 August 2021.
[7] MATLAB for Deep Learning: `https://mathworks.com/solutions/deep-learning.html`, accessed on 4 January 2021.
[8] TensorFlow: `https://www.tensorflow.org/`, accessed on 23 September 2021.

for inference, facilitating the deployment of TensorFlow models on smartphones. It is compatible with both Android and iOS platforms. The Deep Learning Toolbox Converter for ONNX Model Format [9] is a utility, provided by MATLAB, to simplify the transition between MATLAB models and the ONNX format. Import and export functions to and from the ONNX format were introduced in MATLAB R2018a. These functions are compatible with a wide range of DL architectures for image classification, object detection, and more. On the other hand, TensorFlow provides a converter tool that allows the transition from the ONNX model to the TensorFlow format. The TensorFlow Lite Converter [10] is then used to convert the TensorFlow model to the TensorFlow Lite format. See Figure 3.12 for DL frameworks interoperability. The frameworks ecosystem and interoperability contribute to a smoother development process, allowing engineers, scientists, and domain experts to access additional tools and libraries.

3.3.2 Computational Constraints and Solutions

Smartphone's hardware technology has improved to the point where it can now handle some difficult calculations, but it is still in its infancy when it comes to supporting computationally demanding tasks like decision-making and image classification. Additionally, these heavy tasks consume more battery power, making them power-hungry. Thus, a solution is needed to surpass these limitations.

As DL models move to the deployment phase on end-user devices (*e.g.*, IoT devices, smartphones, and tablets), unique computational constraints emerge. Different computing strategies for inference have been proposed to meet application constraints on end-user devices. As shown in Figure 3.13, there are four common computing approaches: (a) on-device computation, (b) cloud-based computation, (c) edge server-based computation, and (d) hybrid computation. We describe in the following each of these strategies.

[9] MATLAB Deep Learning Toolbox Converter for ONNX Model Format; Support package: `https://mathworks.com/matlabcentral/fileexchange/67296-deep-learning-toolbox-converter-for-onnx-model-format`, accessed on 26 August 2021.

[10] TensorFlow Lite Converter: `https://www.tensorflow.org/lite/models/convert/convert_models`, accessed on 23 September 2021.

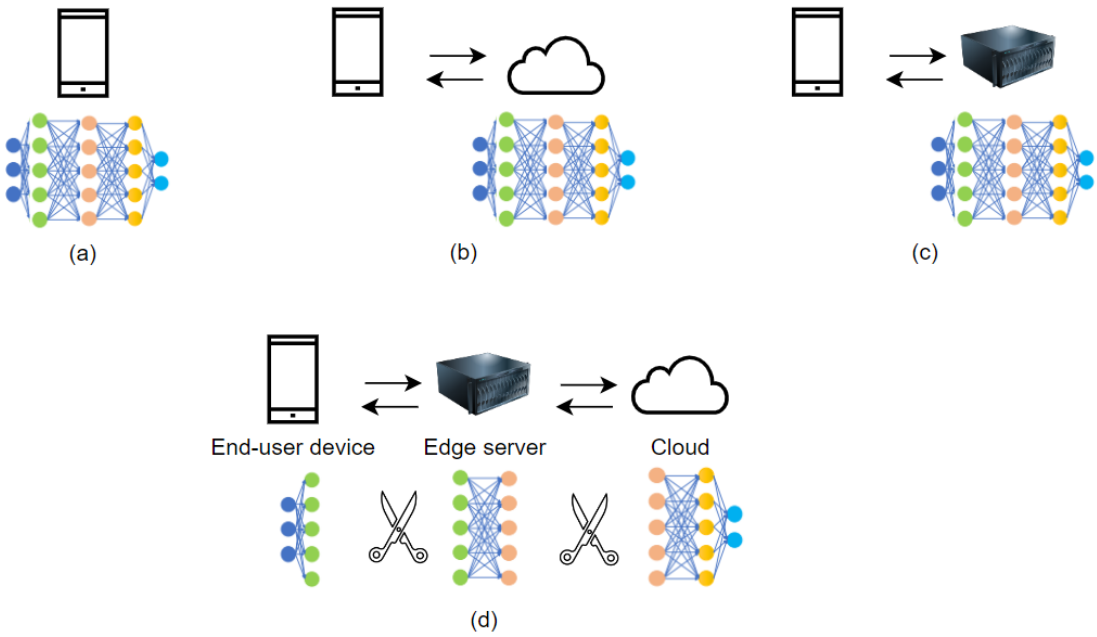


Figure 3.13: (**a**) On-device computation. (**b**) Cloud-based computation. (**c**) Edge server-based computation. (**d**) Hybrid computation.

3.3.2.1 On-device Computation

On-device computation is a strategy in which computational operations are carried out directly on end-user devices. As smartphones are equipped with cameras and other useful sensors, they enable the design and implementation of many beneficial applications. Unfortunately, some applications are heavy to compute, putting forward smartphones constraints: limited processing and computation power, limited battery life in addition to insufficient memory and storage capacity.

3.3.2.2 Server-based Computation

To overcome these limits, the recently-introduced mobile computation offloading can be a solution as it helps in offloading computation from mobile devices to remote cloud servers or local edge servers. Full offloading is mainly the transfer of all computational tasks to a separate processor. As a result, information must be moved from the end-user device, which serves as the data acquisition device, to the server.

Cloud-based computation: Cloud-based computation is leveraged for its processing capabilities and memory capacity as the data is processed on the cloud side, not on the limited-resources devices. It also helps storing the data if required and

thus enables the access to this data later on [Liu19]. Moving computation tasks and storage operations away from the main processor of the smartphone can help save computation time by lowering the running cost of computation-intensive tasks. Basically, using a server for inference keeps the mobile application simple because all the complex tasks are done on the server. Thus, when implementing the system in a mobile user-friendly application, smartphones deficiencies do not have to be considered anymore. This allows unrestricted computing performance and mobility at any moment from any user's device. On another side, limited battery energy impacts the use of the smartphone for heavy tasks as it requires more energy due to high processing requirements, screen use and sensors continuous data acquisition. Given the data, the application must perform a series of tasks requiring specific computation to achieve the desired result. Cloud-based computation can help in saving energy of mobile devices as computation intensive tasks are offloaded to the cloud and can improve reliability by storing and accessing data on cloud side, which decreases the risk of data loss on mobile devices [Liu19]. In addition to these advantages, there are numerous disadvantages [CR19]:

- Bandwidth and scalability: Bandwidth is a main issue in cloud-based computation, and it gets worse with an increasing number of connected mobile devices and data transfer volume. Likewise, as the number of connected devices increases, sending data from the mobile devices to the cloud introduces scalability problems as the cloud entry can become a bottleneck.

- Latency: Cloud-based computation may not always be a good solution when working on real-time applications as data transfer to the cloud may suffer from extra network queuing and transmission delays leading to high latency.

- Service availability: Due to wireless bandwidth limitation, network disconnection and signal attenuation, a connection to the cloud might not always be possible. A sudden internet outage will stop application functionalities as cloud-assisted systems rely on it to transfer data from users' mobile devices to the cloud server and vice versa.

- Privacy: The data sent from end-user devices to the cloud may contain sensitive information leading to privacy concerns. Data sharing and storage in the European Union and the European Economic Area must comply with the

General Data Protection Regulation (GDPR), an EU law regulation on data protection and privacy.

Edge server-based computation: Edge server-based computation can be adopted as a solution to reduce offloading and results in communication delays between mobile devices and the cloud. Instead of offloading tasks to the remote cloud, mobile devices can offload tasks to closer edge servers that help meet required delays by short data transfer intervals. Edge computing is an appropriate solution in case the user or the system cannot wait the time it takes to send the data to a large remote center (cloud) and have results sent back. With computing power at the edge side, decisions and results are received quickly. In addition to the edge servers' power which is higher than end-user devices, these servers conserve network bandwidth usage by doing on-site computing and only sending necessary information for off-site servers. Thus, edge server-based computation helps meet latency, scalability, and also privacy by keeping sensitive data close to the source [CR19, MPI20, WHL+20]. However, there are differing opinions on the safety of edge computing. While some consider edge servers to be a safer option for protecting sensitive data, others believe that data breaches occur more frequently with edge infrastructures due to inadequate security measures. As a result, a robust edge security system must be installed to protect the edge computing infrastructure and ensure its viability. Because of edge computing, less data is sent to the cloud, which may aid in lowering operational costs. On the other hand, the initial investment in hardware and infrastructure for on-premises edge server systems can be considerable. In addition to pricing the server hardware and installing it in a suitable location, infrastructure requires regular maintenance and updating.

3.3.2.3 Hybrid Computation

Mobile computation offloading enables the offloading of parts of the computation tasks (*i.e.*, partial offloading) from mobile devices to remote cloud servers, local edge servers, or both (*i.e.*, edge-cloud computing). This division of computation tasks allows for more efficient use of available resources while still utilizing the processing power of the mobile device. The transfer of information from the end-user device to the server is a key aspect of this approach.

Hybrid computation refers to a combination of different computing approaches that requires the integration of several resources, such as mobile devices, edge

servers, and cloud servers [CR19, MPI20, WHL+20]. As previously mentioned, computation offloading can be full or partial. While the full offloading means that the application is fully executed on the cloud or edge-server side, the partial offloading, which applies to hybrid computing, means that an application is executed on different processing resources. Hybrid computation combines the computational capabilities of mobile devices with the resources of cloud servers and/or edge servers located closer to the devices. In general, lightweight tasks or initial processing are performed on mobile devices, while the more computationally intensive parts are offloaded to the cloud and/or edge server for execution. There are various advantages of using hybrid computation:

- Scalability: Cloud and edge servers offer high-performance computing capabilities, which enables the efficient execution of challenging heavy tasks. The ability to scale resources dynamically based on the workload or demand ensures that the computational requirements of the tasks can be met effectively.
- Network bandwidth: Offloading computationally intensive tasks to servers minimizes the quantity of data that needs to be transferred across the network, which is important when bandwidth is limited. The overall network traffic can be reduced by transmitting only the necessary inputs and receiving only the processed results.
- Latency: Determining which tasks to offload to servers is critical for reducing latency. Offloading computationally complex operations that benefit from server-side processing can increase real-time performance and reduce total response time. Lightweight tasks can be maintained on the mobile device for faster execution.
- Centralized maintenance and updates: By offloading computationally intensive tasks to servers, the server infrastructure carries the main responsibility of maintaining and updating the system. This decreases the complexity and effort necessary for each mobile device maintenance and updates, simplifying overall system management.
- Energy: Hybrid computing architectures can help with energy efficiency. Energy consumption can be lowered through performing lightweight tasks or initial processing on-device. Edge computing lowers the requirement for long-distance data transmission, saving even more energy. Using cloud for

resource-intensive tasks allows for more efficient server infrastructure use and potentially reduced power consumption.

- Privacy: When employing hybrid computing architectures, privacy is a crucial factor, especially when external servers are involved. To guarantee that privacy requirements are respected, task offloading policies should be carefully considered. Offloading only non-sensitive to servers while retaining sensitive data on the mobile device can help to preserve user privacy.

3.4 Conclusion

In this chapter, we investigated the use of built-in smartphone sensors for indoor localization. We provided an in-depth assessment of the capabilities and limits of these sensors in localization. We also provided some examples of existing systems that use smartphone sensors for indoor localization. Furthermore, this chapter demonstrated the interoperability of DL frameworks for smartphone deployment. We also reviewed the benefits and drawbacks associated with the existing computing strategies used in this specific context.

In the next chapters, we will demonstrate two main contributions that we introduce to the field of vision-based indoor localization and sample selection for incremental learning, addressing both practical implementation challenges and the need for robust and efficient solutions in these domains.

Chapter 4

Multi-sensor Data Fusion for Indoor Localization

Sommaire

4.1 Introduction

In previous chapters, we reviewed various elements of the indoor localization task and the current methodologies in the field. These studies show that, with recent progress in computer vision, indoor scene recognition can now be considered a promising room-level localization solution. Indoor scene recognition approaches based on computer vision with DL have led to good results in some situations and environments; however, there is still room for improvement. It is therefore relevant to combine computer vision with other sources of information to overcome this hard problem.

In this chapter, we investigate an approach that takes advantage of smartphone sensors combined with computer vision for indoor room-level positioning. Smartphones are easily accessible devices with built-in cameras that are used on a daily basis. These devices are not only endowed with cameras but also equipped with several built-in sensors that provide the opportunity to acquire additional information and therefore build reliable systems for indoor scene recognition [GWCZ18] as previously stated in Chapter 3. Almost every smartphone has a built-in magnetometer that provides the direction the user is facing, which is known as the magnetic heading. Magnetic heading represents a device's direction relative to the magnetic North. In general, compass heading is the heading measured clockwise from the magnetic North varying from 0° (North) to 360°.

We propose a direction-driven multi-CNN indoor scene localization system based on a combination of image features and the magnetic heading (*i.e.,* pointing direction). We assume that the magnetic heading can be very informative, given that indoor scene recognition constitutes a complex task in computer vision. The proposed system contains four CNNs, each specific to a definite direction range. Given a query image, the system selects the corresponding CNNs for image classification depending on the magnetic heading of the user's smartphone camera. Four CNNs are adopted due to constraints on dataset size. When dealing with a small dataset for training a CNN, two common challenges are overfitting and underfitting. Overfitting occurs when the model becomes excessively tailored to the training dataset, memorizing it instead of generalizing patterns (see Section 2.4.1.2). On the other hand, underfitting occurs when the model is too simplistic to capture the underlying patterns in the dataset. In small datasets, underfitting can occur if the model is too complex relative to the amount of samples available [GJVD18, SP22]. Based

on the above, if we divided the heading directions into more ranges (*i.e.*, the use of more than four CNNs), it would demand a larger number of training images for each range to ensure a robust representation. By choosing four orientation ranges, we can allocate a balanced quantity of images to each range, ensuring that the four CNNs effectively capture distinctive features.

In the training phase, we consider overlapping direction ranges, which means that for a given image direction, the image is fed to two CNNs. This allows for more training images per CNN, since two CNNs may share a subset of the training dataset. In the inference phase, two CNNs are involved in query image classification depending on the magnetic heading of the smartphone camera to obtain more comprehensive features. At the end of this process, to further upgrade the model performance, a weighted fusion method is adopted to determine the final image category and predict the user's specific room location in the indoor space.

We conducted experiments in five different indoor scenes and evaluated classification performance according to accuracy on the whole test set. Compared to the scene recognition method based solely on image features, which is a single-CNN-based classification system fine-tuned on an image training set, the proposed model enables significant improvement in recognition accuracy.

The main contributions of this chapter can be summarized as follows:

- A novel direction-driven architecture of CNNs is introduced to provide an improvement in indoor scene recognition accuracy. Off-the-shelf pretrained CNNs have predefined architectures, with a fixed input size, which prohibits additional information from being provided as an entry. We propose an image classification system guided by supplementary information. The magnetic heading direction of the smartphone assists in vision-based indoor scene recognition, helping the system to identify different specific indoor rooms, taking into account multiple viewpoints.

- A hybrid computing approach is proposed to address latency, scalability, and privacy challenges. In general, meeting the computational requirements of DL with the limited resources of handheld devices is not feasible. Several works have combined on-device computing with edge computing and/or cloud computing, resulting in hybrid architectures [CR19]. We take advantage of these new computing techniques to propose a global system computing strategy that meets users' needs.

- While several indoor and/or outdoor localization datasets exist in the literature, none of them integrates information other than images. To overcome this issue, we provide a dataset containing images with their respective magnetic heading direction in the metadata.

The rest of this chapter is organized as follows. Section 4.2 describes the investigated indoor scene recognition approach, as well as each component of its architecture. Section 4.3 discusses the partitioning of the proposed DL model. Section 4.4 depicts the different experiments conducted on a real dataset, in addition to the effect of model partitioning on system computation. Finally, Section 4.5 concludes the chapter with a discussion of future work.

4.2 Direction-driven Convolutional Neural Networks for Indoor Scene Recognition

Our primary focus in this thesis lies in directly using built-in smartphone sensors. To overcome the difficulties encountered with vision-based indoor localization, researchers have proposed the integration of other sensors. Today, all smartphones are equipped with a magnetometer, making it a universally accessible sensor for indoor localization. The magnetometer is designed to measure the Earth's magnetic field. In indoor environments, where GNSS signals are unreliable, and WiFi and Bluetooth signals can be obstructed by walls and structures, the Earth's magnetic field remains relatively stable. Magnetometer can complement other smartphone sensors for indoor localization, offering a cost-effective solution. Our work aligns with the concept of fusion of visual and magnetic sensor data. We propose to use built-in smartphone sensors to provide valuable information about the device's magnetic heading with respect to the magnetic North (Section 4.2.1) and then investigate it in the proposed localization system (Section 4.2.2).

4.2.1 Magnetic Heading Estimation

The smartphone sensors' outputs are provided with respect to the smartphone's reference frame. The smartphone reference frame is defined by the X_s, Y_s, and Z_s axes. These axes are oriented relative to the smartphone terminal screen, with its center serving as the origin of the smartphone reference frame. The X_s axis points to the

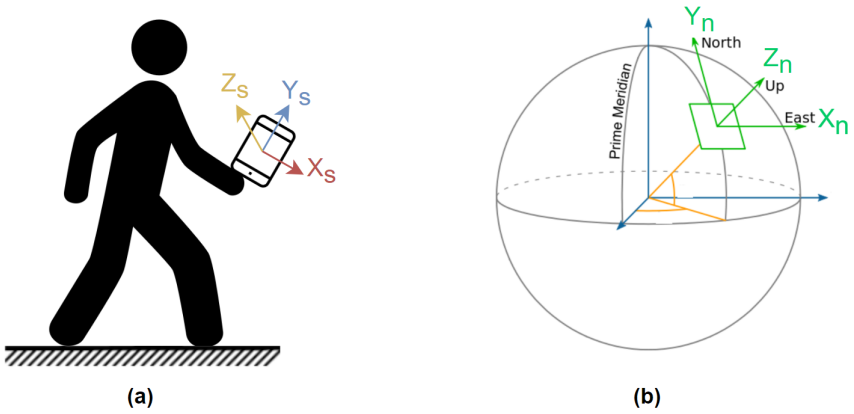


Figure 4.1: **(a)** The smartphone coordinate system. **(b)** The navigation coordinate system.

right (in the horizontal direction), the Y_s axis points to the front (in the vertical direction), and the Z_s axis points upward (in the outward direction) from the screen, as represented in Figure 4.1a. However, in order to align the smartphone actions with the surrounding world, a connection with the navigation coordinate system is required. The navigation coordinate system usually follows the Earth's surface, with the point of contact representing its origin. The X_n axis points to the East, the Y_n axis points to the North, and the Z_n axis points up, as illustrated in Figure 4.1b.

In our application, the process entails orienting the device vertically with the user pointing the camera in the desired direction. In the context of heading estimation, collected sensors data from both the magnetometer and the accelerometer are used to bridge the gap between the two coordinate systems and calculate the smartphone's heading as described in the following. The magnetometer is designed to measure the Earth's magnetic field in three dimensions (X_s, Y_s, and Z_s). This sensor provides data related to the strength and direction of the magnetic field in the smartphone's vicinity. To enhance the accuracy of determining the smartphone's orientation and heading, this magnetic field data is combined with information from the accelerometer. The accelerometer is responsible for sensing gravity to give the smartphone tilt information. It is imperative to maintain the assumption that the accelerometer primarily detects the constant force of gravity while remaining static in orientation. By combining the data from the magnetometer and the accelerometer, the smartphone's software can precisely determine the smartphone's heading with respect to the magnetic North. The accelerometer and magnetometer sensors operate in a continuous data reporting mode, generating high-frequency data updates. These sensors are well-suited for real-time applications, such as ori-

entation tracking and motion sensing.

To get the smartphone's orientation angles, first, a rotation matrix *Rot* is obtained using the accelerometer and the magnetometer data. The rotation matrix describes how the smartphone is oriented to the Earth's navigation reference frame. *Rot* is a 3×3 matrix that describes the transformation required to bring the smartphone's coordinate system to the navigation coordinate system.

In Android, it is obtained using the accelerometer and the magnetometer data (represented as *A* and *M*, respectively) as follows:

$$\textit{SensorManager.getRotationMatrix(Rot, I, A, M)}, \tag{4.1}$$

with *I* being a 3×3 matrix that represents the inclination matrix. It describes how much the Earth's magnetic field is inclined with respect to the gravity vector.

Then, remapping is applied to the original rotation matrix *Rot*. The remapping of the rotation matrix guarantees that the smartphone coordinate system is aligned with the plane parallel to the Earth's surface, allowing an accurate calculation of the smartphone's tilt angles with respect to that plane. The orientation *O* corresponds to an array of length 3 and contains the orientation angles of the smartphone around X_s, Y_s, and Z_s axes. *O* is represented as follows

$$O = \begin{bmatrix} o_0 \\ o_1 \\ o_2 \end{bmatrix}. \tag{4.2}$$

In Android, the orientation *O* is obtained using the rotation matrix *Rot* as follows

$$\textit{SensorManager.getOrientation(Rot, O)}, \tag{4.3}$$

with the orientation angles in radians.

The magnetic heading of the user's smartphone camera with respect to the magnetic North is defined as o_0, which can be written in degree as

$$\theta = o_0 \times \frac{180}{\pi}. \tag{4.4}$$

4.2.2 Localization System Architecture

Knowing that indoor scenes are very complex due to strong change in viewpoints and high similarity between scenes, additional information could be of great interest. Our intuition relies on the assumption that the camera heading of the user smartphone relative to the magnetic North can be very informative. It informs the image classification system as to which way the smartphone's camera is facing. We combine accelerometer information with magnetometer data for magnetic heading estimation, which allows for correct orientation relative to North when the smartphone is held vertically (*i.e.*, determining the camera facing when capturing an image).

We consider a direction-driven multi-CNN system for indoor scene recognition that takes into account the magnetic heading of the user's smartphone (θ). The global architecture illustrated in Figure 4.2 consists of three main components: the selection block for direction-driven model selection, the image classification model defining four CNN models, and the fusion and decision block for combination of the obtained results. These three components are described in detail in Sections 4.2.2.1–4.2.2.3, respectively. In order to cover the four ranges of orientations (A, B, C, and D in Figure 4.3a), the proposed classification system contains four CNNs, denoted as A, B, C, and D. The use of four ranges of orientations is motivated by small datasets and the need to avoid underfitting or overfitting. Dividing the heading directions into more ranges would necessitate a greater number of images for each range to provide adequate representation. This strategy allows us to assign a fair number of images to each range, guaranteeing that the CNNs can efficiently learn distinguishing features. Algorithm 1 presents the process followed for indoor scene image classification in the inference phase.

In the following, we describe in detail the three main building blocks of the proposed method.

4.2.2.1 Selection Block

The main objective of the selection block is to select two of the four available CNNs in order to use them for indoor scene recognition. For training or inference, given an image, two CNNs are selected according to the quadrant to which the magnetic heading (θ) of the camera belongs. More precisely, as in Figure 4.3a, the selection rule is as follows:

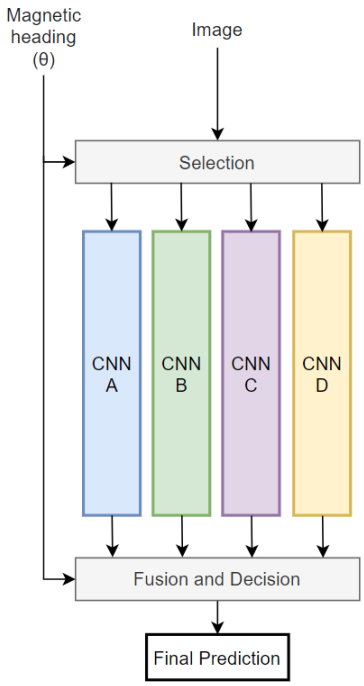


Figure 4.2: Architecture of the proposed system with four CNNs.

- Between North and East: select CNN A and CNN B;
- Between East and South: select CNN B and CNN C;
- Between South and West: select CNN C and CNN D;
- Between West and North: select CNN D and CNN A.

The outputs of the two selected CNNs are subject to weighted fusion performed in the fusion and decision block, as described in Section 4.2.2.3. Before, we provide a detailed description of the image classification models.

4.2.2.2 Image Classification Models

We propose a generic system that can include any type of deep neural network used for image classification, even though we focus on CNNs. Pretrained CNNs are trained on more than a million images from the ImageNet dataset [DDS+09, KSH12]. Consequently, these networks learn rich feature representations from a wide range of images. Instead of building CNN models from scratch, we investigated mobile-compatible pretrained CNNs, namely SqueezeNet [IHM+16], ShuffleNet [ZZLS18], and MobileNet [HZC+17, SHZ+18] (see Section 2.4.1.5). These lightweight CNN

Algorithm 1 Inference classification methodology

Input: Query image, Magnetic heading θ.

1: Determine the quadrant to which the magnetic heading θ belongs
2: Select the two corresponding CNNs according to the selection rule defined in Section 4.2.2.1
3: p_1 = Estimated probabilities with the first selected CNN
4: p_2 = Estimated probabilities with the second selected CNN
5: $\alpha(\theta')$ = Weighting parameter of the fusion method with (4.7) or (4.8)
6: $p = \alpha(\theta')\ p_1 + \left(1 - \alpha(\theta')\right)\ p_2$
7: Predict the image category with $\max(p)$

Output: Prediction of the specific indoor room.

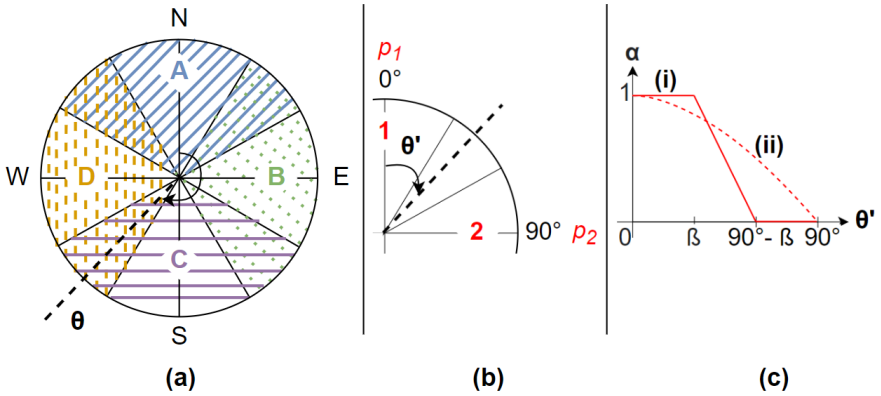


Figure 4.3: (**a**) CNN selection depending on the magnetic heading of the image (θ). (**b**) Weighted fusion strategy. (**c**) Fusion techniques: (**i**) piecewise linear and (**ii**) cosinusoidal.

models have demonstrated a good tradeoff between accuracy and efficiency while addressing resource-constrained environments, including low memory and hardware requirements.

Transfer learning using pretrained CNNs on ImageNet is important as these models have already learned to extract low-level and mid-level features, making them a valuable resource for a variety of computer vision tasks. Despite the ImageNet dataset focuses on objects, some features, such as edges and textures, are common between objects and indoor scenes. Moreover, limited indoor scenes data can hinder the ability of CNNs to generalize well and may lead to overfitting. Pretrained models reduce the need for extensive labeled data (see Sections 2.4.1.2 and 2.4.1.3). By transfer learning, we fine-tuned these pretrained CNN models as follows. The top layer (fully connected layer or convolutional layer) of the pretrained CNN is replaced with a new layer having a number of outputs equal to the number of categories in the target dataset. A softmax activation function is introduced at the

output of the CNN with a number of neurons equals the number of categories to obtain a probability vector as an output. A well-known technique in transfer learning consists of freezing some trainable layers. The weights of those frozen layers are not updated during fine tuning. In general, the frozen layers are selected from the first convolutional layers of the model because the last convolutional layers are more task-specific; therefore, applying fine-tuning to these layers is important to enhance learning quality. Moreover, freezing the weights of several layers can significantly speed up network training.

4.2.2.3 Fusion and Decision Block

As mentioned in Section 4.2.2.1, two CNNs are selected based on the quadrant to which the magnetic heading (θ) of the image belongs. Therefore, a weighted fusion technique is applied to the two probability vectors (p_1 and p_2) corresponding to the inference outputs of the two selected CNNs. The adopted principle in the fusion block is that, when classifying an indoor scene query image, each of the two selected paths contributes to the final decision by a factor depending on the value of θ.

In order to provide a single formulation for all four possible quadrants shown in Figure 4.3a, we represent them in a single quadrant using the modulo operation as follows

$$\theta' = \theta \mod 90^\circ, \tag{4.5}$$

namely the modified magnetic heading of the smartphone camera ($\theta' \in [0^\circ, 90^\circ[$). As depicted in Figure 4.3b, p_1 corresponds to the probability vector at the output of the specific CNN for the range of θ' centered at the vertical axis and p_2 at the output of the CNN whose specific range of θ' is centered at the horizontal axis. Thus, the proposed fusion method is defined as

$$p = \alpha(\theta')\ p_1 + \big(1 - \alpha(\theta')\big)\ p_2, \tag{4.6}$$

where $\alpha(\theta')$ is the weighting parameter calculated to combine the two probability vectors (p_1 and p_2) as described in Algorithm 1.

We consider two weighted fusion strategies based on the smartphone's magnetic heading. The first strategy is the piecewise linear weighted fusion, as represented in Figure 4.3c(i). Inspired by fuzzy logic, let β be the hyperparameter defining the different intervals of weighting that can take values in the range of $[0^\circ, 90^\circ[$. For this

first fusion method, the weighting parameter $\alpha(\theta')$ is defined as

$$\alpha(\theta') = \begin{cases} 1 & \text{if} \quad \theta' \in [0^\circ, \beta[\\ \frac{1}{2\beta - 90^\circ}\,\theta' + \frac{\beta - 90^\circ}{2\beta - 90^\circ} & \text{if} \quad \theta' \in [\beta, 90^\circ - \beta] \\ 0 & \text{if} \quad \theta' \in]90^\circ - \beta, 90^\circ[\end{cases} \tag{4.7}$$

with a special case when $\beta = 45^\circ$. In this case,

$$\alpha(\theta') = \frac{1}{2} \quad \text{if} \quad \theta' = \beta. \tag{4.7a}$$

In this chapter, we deal with the following three cases of linear weighted fusion: $\beta = 0^\circ$, $\beta = 30^\circ$, and $\beta = 45^\circ$. The second fusion strategy is the cosinusoidal weighted fusion, as illustrated in Figure 4.3c(ii), with $\alpha(\theta')$ defined as

$$\alpha(\theta') = \cos(\theta') \quad \forall \theta' \in [0^\circ, 90^\circ[. \tag{4.8}$$

After applying one of the fusion techniques, the category with the maximum classification probability is selected, namely

$$\max(p). \tag{4.9}$$

This leads to the final prediction of the specific indoor room.

4.3 Global System Architecture

End-user devices are used to generate and collect data that often necessitate real-time analysis through DL models. These valuable data also serve as an input to train DL models. However, the efficient execution of DL inference and training poses a significant computational challenge, requiring substantial resources. In this section, we discuss distributed DL training, as well as DL task partitioning, which are fundamental approaches that optimize training workflows and enhance model performance. We propose a hybrid computing approach to address latency, scalability, and privacy issues.

4.3.1 Distributed Deep Learning Training

DL model training is a time-consuming process. To speed up this process, distributed training can be used. Distributed training can be accomplished through data parallelism or model parallelism. These techniques leverage multiple processors that work in parallel. Data parallelism is one method of parallelism that involves splitting the training data and processing each split on a separate processor in parallel. Each processor independently trains a copy of the DL model on a different subset of the training data [SLA+18, LHRX20]. Data parallelism can be in synchronous or asynchronous mode. In synchronous mode, all processors handle different training data splits and aggregate the models' weights and gradients at the end of each processor computation. On the other hand, in asynchronous mode, each processor works independently on its training data split and updates the model's weights without waiting for other processors. This asynchronous process allows each processor to update the model at its own pace [LZV+20, AKSM22]. Generally, data parallelism is a straightforward and efficient method for accelerating DL model training. On the other hand, model parallelism is used when a DL model is extremely large to fit into the memory of a single device or processor. In this case, the model is divided into submodels, and each submodel is loaded onto a separate processor. During training, each processor processes its allocated part of the model. Like data parallelism, communication is required to synchronize the model's weights and gradients among the processors [LHRX20].

Distributed DL training gains popularity and proves effective in situations where the computational requirements for training deep neural networks exceed the capacity of a single processor or when rapid and efficient convergence is required [LHRX20, BSD+21]. The training process is distributed depending on the available computation resources. In the proposed indoor scene recognition system, each of the four CNNs is specific for a predetermined range of magnetic headings (see Section 4.2.2.1). Thus, the four CNNs can be trained in parallel. Each CNN is trained on its training data subset on a separate processor using data parallelism [1], allowing for simultaneous training. Once all CNNs are trained, their predictions are combined for inference with the fusion and decision block (see Section 4.2.2.3). Through weighted fusion, aggregation harnesses the strengths of each CNN. The inherent scalability of our multi-CNN approach is one of its key benefits, closely

[1] MATLAB Parallel Computing Toolbox: `https://mathworks.com/products/parallel-computing.html`, accessed on 20 November 2023.

following the ideas of data parallelism. Our approach effortlessly scales to accept larger datasets and speed up processing as processing units become available. Data parallelism gives the proposed system the ability to effectively handle data without compromising performance, making it suitable for a variety of applications and situations.

4.3.2 Computing and Partitioning Deep Learning Tasks

The DL inference can be performed on cloud servers, referred to as cloud-based deep inference, or on edge servers, referred to as edge-server-based deep inference. Alternatively, inference can be performed locally using mobile CPU and GPU, referred as on-device deep inference [CR19].

The DL inference is computationally intensive. Even after the creation of lightweight mobile-compatible deep neural networks, smartphones remain clearly inferior to edge and cloud servers in terms of performance, as several deep neural networks may be needed during inference in some applications. In the case of image classification, the inference computational demands of DL are strongly reliant on the increase in computing power. In general, meeting the computational requirements of DL necessitates cloud computing, as it guarantees limitless on-demand processing power.

Recent studies have shown that splitting the network between the mobile device and cloud and/or edge servers can improve the end-to-end latency of deep neural networks inference [WHL+20, MMH+21]. One way of using hybrid computing with partial offloading with DL models is CNN model partitioning [KHG+17, XZC+19]. In such approaches, instead of creating an application handling everything, CNN architectures are distributed between the mobile device and cloud and/or edge server, as shown in Figure 3.13d. Thus, some layers are computed on the mobile device while other layers are computed by the cloud and/or the edge server, which may reduce the computation power required on the smartphone. The key aspect when distributing computing between the mobile device and the cloud and/or edge server is which data must be stored and processed locally. The optimal computation partitions for offloading are difficult to choose, requiring a separate study and analysis.

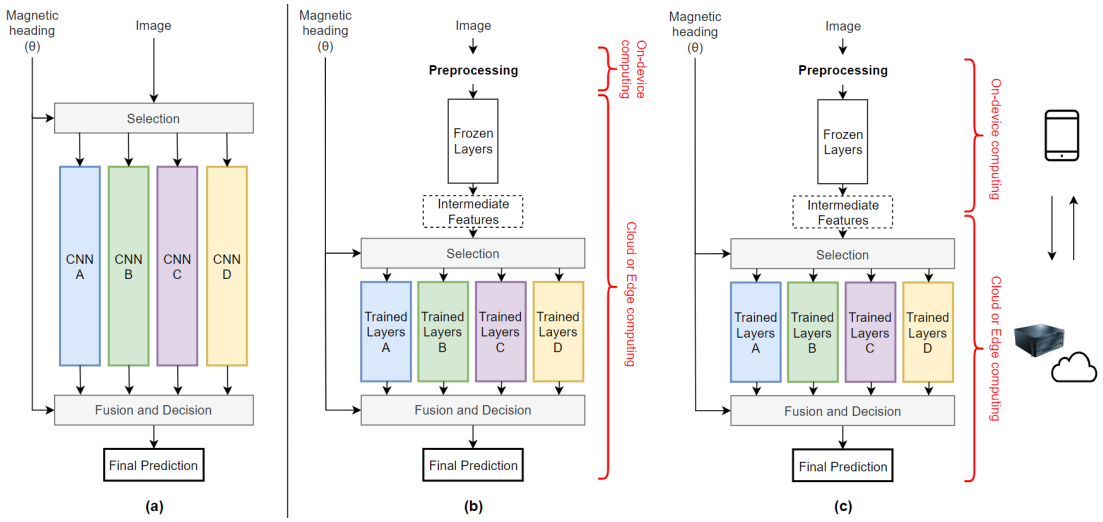


Figure 4.4: (**a**) Architecture of the proposed system with four CNNs. (**b**) Computing strategy with full offloading, with the four CNNs partitioned in the common submodel (*i.e.,* frozen layers) and the other four submodels (*i.e.,* trained layers A, B, C, and D). (**c**) Computing strategy with partial offloading.

4.3.3 Partitioning of the Proposed Model

DL model partitioning is the process of dividing a DL model into multiple parts that can each be deployed and run on different computing devices and servers. In the proposed indoor scene recognition system, several CNN models need to be saved and used during inference, requiring an increase in memory capacity and computation power. Including all the needed models in the mobile application bundle also significantly increases its size, up to many megabytes (MB) in practice. The proposed direction-driven CNNs model has a common inference part because some of the layers of the CNNs are frozen during the training phase. Thus, the architecture of the proposed system represented in Figure 4.4a is partitioned into five submodels: the common submodel (*i.e.,* frozen layers) and the other four submodels (*i.e.,* trained layers A, B, C, and D).

Two computing strategies are considered; Figure 4.4b represents full offloading (*i.e.,* cloud-based computing or edge-server-based computing), while Figure 4.4c constitutes partial offloading (*i.e.,* hybrid computing). In the case of full offloading, the captured image is preprocessed and sent to the cloud or edge server for full computation. In the case of partial offloading, the common submodel is computed on the user's end device (making preliminary predictions before sending the data), and the output intermediate features are sent to the server. Then, the other

four submodels are computed on a cloud or edge server. For these two computing strategies, the final prediction (*i.e.,* user's room-level position in the indoor environment) is sent back from the server to the user side.

The primary goal is to minimize the end-to-end latency while respecting end-user devices and server constraints (see Section 3.3.2). It is important to note that the practical implementation and testing of an application on a smartphone, as well as the connection with a server to assess the most appropriate computation approach, were not within the scope of this research. We concentrated our efforts during the study on building and analyzing the proposed indoor localization system employing built-in smartphone sensors and DL. However, we realize the vital necessity of these practical issues in fulfilling our proposed framework's full potential. Partitioning into submodels is based on the communication and computational costs of the submodels; thus, it depends mainly on the layer types, the per-layer output size (*i.e.,* activation), the per-layer data communication latency, the per-layer computation latency (*i.e.,* server and end-user devices processing latency), and the memory footprint. As described in the following section, we conducted experiments in order to provide a deep insight into the explored partitioning of the direction-driven model.

4.4 Experiments and Results

4.4.1 Dataset Preparation

To construct and evaluate the proposed indoor scene classification system, we first created a dataset of images with their respective magnetic headings with respect to the magnetic North. The prepared dataset includes informative images of the indoor environment with different perspectives of the studied rooms.

To ensure an efficient data collection process, we designed an Android application that uses the smartphone's built-in sensors. When capturing images using this application, each image is collected with the corresponding magnetic heading saved in its metadata (see Section 4.2.1). For data collection, the smartphone was held in the portrait/vertical position. The RGB images were cropped and saved at a size of 1088×1088 pixels to avoid distorting the shapes of the objects in the images when resized. The dataset was prepared using the main rear camera of a Samsung Galaxy A51.

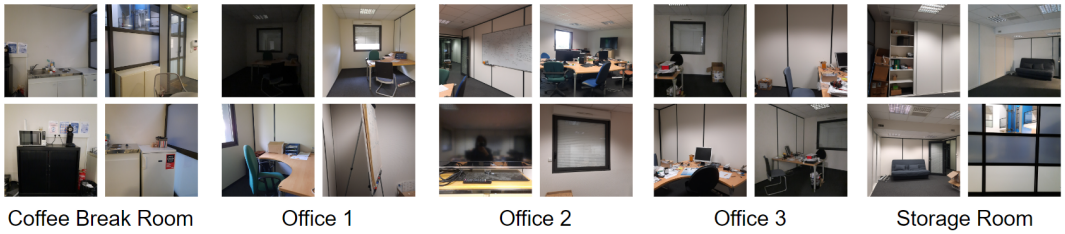


Figure 4.5: Examples from the different classes of the collected dataset.

The indoor environment studied in this work has five rooms: coffee break room, office 1, office 2, office 3, and storage room. To provide diverse and representative data, the data collection process was conducted over several days. We took precautions to maintain consistency during the data collection period. Two data collection rounds were conducted. In the first round, we took eight images per position (*i.e.,* a given standing location) in each room with different orientations. Each position used to collect images results in a distinct perspective. Images were collected at orientations of 0° (North), 45°, 90°, 135°, 180°, 225°, 270°, and 315°. These images were used for training the proposed direction-driven multi-CNN system. In the second round, we took an average of 20 images per position in each room, with different positions than the first round and a full 360-degree rotation in each position to take all the heading viewpoints into consideration. The entire dataset was then cleaned by deleting uninformative images (*i.e.,* images constituted mostly of walls, windows, etc.). We obtained between 100 and 200 images per class depending on the room dimensions and complexity. A part of these collected images was used for training and validation of the system, and the rest were used for assessment of the classification accuracy (50% for the training phase and 50% for the testing phase). Figure 4.5 shows some examples from the collected dataset.

4.4.2 CNN Training and System Testing

For the proposed indoor scene recognition system based on the direction-driven multi-CNN architecture, the four CNN models need to be trained and validated in order to be implemented. To assess its performance, the proposed classification system was evaluated on the totality of the testing set. We also used the testing set to examine the relevance of the fusion strategies described in Section 4.2.2.3.

As previously mentioned, since we had few real data points, we relied on CNN models pretrained on ImageNet [DDS+09, KSH12], and fine-tuned them using the

real dataset. We examined three well-known mobile-compatible pretrained CNNs: SqueezeNet [IHM+16], ShuffleNet [ZZLS18], and MobileNet version 2 [SHZ+18]. In order to provide a baseline system, we trained and fine-tuned a single CNN model with the totality of the training and validation sets. In order to provide a fair comparative analysis, we considered the same pretrained CNN used for the proposed recognition system.

Models were optimized using a batch gradient descent optimizer with a learning rate equal to 0.001. Note that pretrained CNNs take fixed image sizes and a defined number of input channels; therefore, all the images in the dataset were preprocessed (Images from our real dataset were scaled to $224 \times 224 \times 3$ to respect the dimensions accepted by the input layer of the pretrained CNNs ShuffleNet and MobileNet and to $227 \times 227 \times 3$ when working with SqueezeNet). We trained the models for a maximum of 500 epochs. In order to avoid overfitting, the CNN training stopped automatically when the validation loss starts increasing while the training loss was still decreasing. Simulations were implemented using MATLAB R2019a.

4.4.3 Performance Evaluation

We computed the standard performance metric for image classification to assess performance. The test accuracy is defined as

$$\text{Accuracy} = \frac{\text{Total number of test images correctly classified}}{\text{Total number of test images}}.$$

Five Monte Carlo simulations were conducted to evaluate our direction-driven multi-CNN model, as well as the single-CNN baseline system. The average test accuracies, denoted as Accuracy_{avg}, are presented in Table 4.5 for the three pretrained CNN models and the fusion strategies. The proposed indoor scene recognition approach outperformed the baseline system in terms of accuracy for all proposed fusion strategies. The results show that the linear weighted fusion with $\beta = 0°$ achieved the best performance, proving the necessity of combining the two selected CNNs.

Additional tests were carried out to evaluate the performance of various selection rules and their impact on the overall system accuracy. We investigated several scenarios, including one in which the system selects the quadrant that is completely opposite to the magnetic heading, resulting in the selection of the two opposite CNNs. For example, if the magnetic heading is between North and East, select CNN C and CNN D (*i.e.,* the complete opposite of the proposed selection block approach,

Table 4.1: Comparison of the accuracy ($_{avg}$(%)) between the baseline system and the proposed approach (best results are in bold).

Pre-trained Model	Baseline	Proposed Approach			
		Linear Fusion			Cosinusoidal Fusion
		β = 0°	β = 30°	β = 45°	
SqueezeNet	67.52 ± 1.95	**81.02 ± 2.75**	77.50 ± 3.30	77.02 ± 3.30	79.52 ± 2.62
ShuffleNet	88.98 ± 2.03	**92.22 ± 0.41**	91.40 ± 0.54	91.34 ± 0.44	91.94 ± 0.69
MobileNet-v2	90.66 ± 1.80	**93.10 ± 0.56**	92.62 ± 1.03	92.50 ± 0.82	92.44 ± 0.82

Table 4.2: Comparison of the accuracy ($_{avg}$(%)) between the baseline system and different selection rules (best results are in bold).

Pre-trained Model	Baseline	Different Selection Rules		
		Opposite Quadrant Selection with Linear Fusion (β = 0°)	One Random CNN	One Specific CNN
SqueezeNet	**67.52**	32.96	52.96	51.91
ShuffleNet	**88.98**	44.86	63.38	63.45
MobileNet-v2	**90.66**	46.30	63.92	64.21

which selects CNN A and CNN B in this case). We also assessed the system's performance when only one CNN was chosen randomly from the four available CNNs, as well as when one specific CNN was chosen for all test images. Table 4.2 shows the results of these tests, which demonstrate how including these alternate scenarios considerably affects the system's accuracy. The performance in these three scenarios falls significantly short of what the proposed approach presented in Algorithm 1 achieves. In terms of accuracy, the system performs worse than the baseline system when using the opposite two CNNs to the corresponding magnetic heading quadrant for fusion or when omitting the fusion block and instead depending on one of the four available CNNs. These findings highlight the importance and efficacy of the proposed approach, demonstrating higher accuracy. The proposed selection block and fusion block are crucial in improving the system's performance, resulting in increased accuracy for indoor scene recognition.

4.4.4 Stability Analysis: Effect of Sensor Accuracy on the System

As explained in Section 3.2.4.2, there are several challenges that can affect the accurate estimation of the magnetic heading, impacting the overall performance of the proposed localization system. In [HNO13], a sensor accuracy test was conducted in a large industrial hall with seven different mobile devices with non-identical built-

in sensors, analyzing the impact of a harsh environment and different hardware on smartphones' digital compass estimation. At each position in the studied environment, the divergence of the magnetic heading provided by the smartphones from the correct heading from the construction plan was recorded. Examination of the collected device measurements showed that, for most mobile devices, the probability of having a magnetic heading error below 20° is around 85%. In [NP17], over 14,000 readings from two Android devices were collected to evaluate the stability of the sensors' readings. Smartphone Honor 3C provided quite consistent sensor data with a heading error of about ±2°. Lenovo B8080-F tablet provided a lower sensor quality with a heading error of about ±20°.

To analyze the effect of the error when estimating the magnetic heading on the proposed system's performance and stability, we conducted magnetic heading error simulations. We assumed that the magnetic heading error, denoted as e, follows a normal distribution:

$$e \sim \mathcal{N}(\mu, \sigma^2). \tag{4.10}$$

Thus, the normal probability density is guided by the mean (μ) and the standard deviation (σ) and is defined as follows:

$$P(e) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(e-\mu)^2}{2\sigma^2}\right). \tag{4.11}$$

We computed the commonly used image classification performance measure to assess the stability of the proposed system. This time, the accuracy can be measured as

$$\theta_e = (\theta + e) \mod 360°, \tag{4.12}$$

where θ is the magnetic heading of the smartphone's camera when the image is captured, and e is the random Gaussian error as previously defined. The expression of the modified magnetic heading (4.5) leads to

$$\theta' = \theta_e \mod 90°. \tag{4.13}$$

Based on [HNO13], knowing that we are not working in an industrial indoor environment and that smartphones have come a long way over the past few years, we simulated error values following a normal distribution with several values of σ and $\mu = 0°$. The obtained results are represented in Table 4.3. The results show a performance reduction when simulating an error following a normal distribution with

Table 4.3: Comparison of the accuracy ($_{avg}$(%)) between the baseline system and the proposed approach with the linear fusion ($\beta = 0°$) and simulating error on magnetic heading (best results are in bold and worst results are highlighted in red).

Pre-trained Model	Baseline	Proposed Approach with Linear Fusion ($\beta = 0°$)				
		$e = 0$	$\sigma = 30$	$\sigma = 60$	$\sigma = 90$	$\sigma = 120$
SqueezeNet	67.52	**81.02**	80.18	79.26	77.84	75.78
ShuffleNet	88.98	**92.22**	91.46	91.24	89.12	88.10
MobileNet-v2	90.66	**93.10**	92.86	92.60	91.30	89.60

Table 4.4: Model file size of the different implementations with MobileNet-v2.

Framework	Model	Model File Size
ONNX	Four Complete CNNs	35 MB
	Proposed Computing Strategy	19 MB

$\mu = 0°$ and $\sigma = \{30°, 60°, 90°\}$ on the proposed indoor recognition model using linear weighted fusion with $\beta = 0°$. Nonetheless, the proposed model still outperforms the baseline system, demonstrating the relevance of our approach. An error following a normal distribution with $\mu = 0°$ and $\sigma = 120°$ causes a drop in accuracy, resulting in the worst performance when compared to the baseline. However, such a high value of variance error is not practical in general [HNO13, NP17].

4.4.5 Model Analysis and Partitioning for Inference

In a CNN model, each layer has its own set of learnable weights that are optimized during training by minimizing the classification loss. These parameters are typically saved in a model file that can be loaded into memory during inference. ONNX [2] is an important DL model format because it provides a common standard for representing DL models, making it easier to develop and deploy them across multiple frameworks and devices (see Section 3.3.1). We compared the model file size of each implementation (with MobileNet-v2) as shown in Table 4.4. The proposed computing strategy (composed of frozen layers and trained layers) is lighter than the four complete direction-driven CNNs by about 16 MB.

Additionally, Figure 4.6 describes the per-layer output data size (in MB) for the four complete CNNs represented in Figure 4.4a compared to the proposed computing strategies as in Figure 4.4b, and c. We can observe the following. First, adopt-

[2] Open Neural Network Exchange; GitHub repository: `https://github.com/onnx/onnx`, accessed on 26 January 2022.

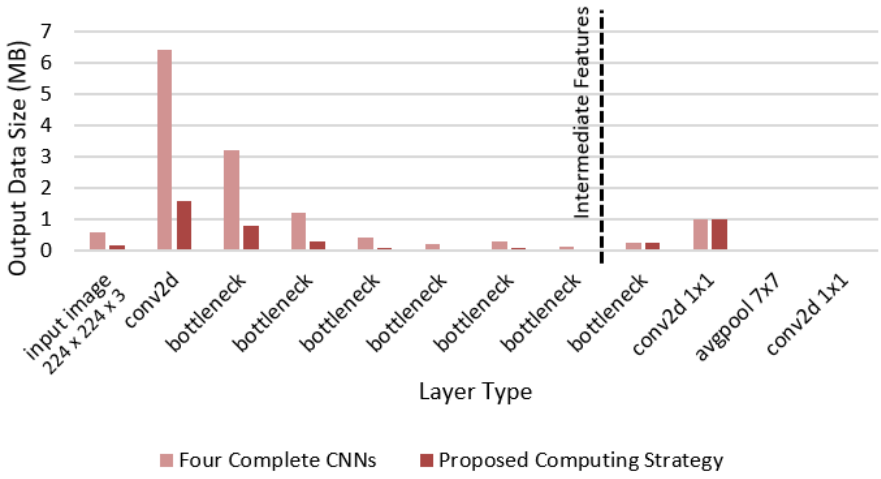


Figure 4.6: Per-layer output data size (MB) for four complete CNNs compared to the proposed computing strategy with MobileNet-v2. The dotted vertical line is the split point based on on-device (frozen layers) and cloud/edge-server computing as in Figure 4.4.

Table 4.5: Proposed computing strategy submodel sizes and outputs with MobileNet-v2 based on partial offloading as in Figure 4.4c.

Submodels	Submodel File Size (MB)	Output Data Size (MB)	Computing Strategy
Frozen Layers	5.31	3.136×10^{-2}	On-Device (User-Side)
A (Trained Layers)	3.44×4	0.002×10^{-2}	Cloud or Edge Server
B (Trained Layers)			
C (Trained Layers)			
D (Trained Layers)			

ing the proposed computing strategy is better than using the four complete CNNs because the common frozen layers are implemented once rather than four times, resulting in less computation time and required power. Second, because the input image size is larger than the intermediate feature size, splitting the CNN into two parts (*i.e.*, a first part running on the mobile device and a second part running on a cloud or edge server) may be more beneficial. As a result, the submodels are deployed in the manner described in Table 4.5 (*i.e.*, partial offloading as in Figure 4.4c).

4.5 Conclusion

In this chapter, we proposed a direction-driven multi-CNN indoor scene recognition system for room-level localization that uses embedded smartphone sensors to account for the camera heading direction relative to magnetic North. We created a

dataset that includes images with corresponding magnetic heading values. We also used and compared two heading-based weighted fusion techniques. Experiments showed that the proposed system outperforms the baseline system based solely on images. When dealing with indoor scene image data, the proposed system outperformed the traditional CNN image classification system. The proposed system relies on built-in smartphone sensors, which vary in quality and accuracy across different devices and environments and may have an impact on the overall performance of the system. We also investigated how magnetic heading error affected the proposed system, demonstrating the utility and stability of our method. Additionally, we discussed the current computing paradigms and how they apply to DL tasks. We also analyzed the effect of model partitioning on system computation and proposed a hybrid computing strategy for our scene recognition system (*i.e.,* model partial offloading between the mobile device and server).

In the next chapter, we will focus on the applicability of the proposed system in terms of maintenance when additional new rooms must be integrated and identified, requiring the system to be deployed over a larger area.

Chapter 5

Coherence-based Sample Selection for Class-incremental Learning

Sommaire

5.1 Introduction

Maintenance is a critical step for a vision-based scene recognition system, as it allows the system to evolve and accommodate new specific scenes over time. The localization system may be delivered with a set of preliminary indoor scene recognition capabilities, but new rooms may need to be added with time. Additionally, in order for embedded devices and platforms to operate effectively in real-world situations, computational and memory efficiency are essential. Thus, memory-controlled incremental approaches are required.

CNNs have been established as a cornerstone in computer vision. They have exhibited remarkable achievements in image classification, object detection, and semantic segmentation. While CNNs have excelled at several vision tasks, they encounter challenges when it comes to continuous knowledge learning, specifically in scenarios involving incremental learning of new classes. For example, in the context of indoor scene recognition systems, there is a necessity for incremental learning to assimilate new classes over time. As a result, the learning system must possess the capability to assimilate novel knowledge persistently to remain effective in evolving scenarios. Due to these requirements, the Class-IL domain has been recently advanced, aiming to continually build a classifier that contains all encountered classes [ZWQ+23].

Class-IL involves incrementally updating the recognition system at each time, by adding new classes from a new training dataset (also called task) available at some time [HWC22]. As a sub-field of Incremental Learning (IL) within the Continual Learning paradigm, it should accommodate the memory limit by not storing all training data and learning from scratch at each time. Due to storage limitation, the model is only able to access the dataset at the current task. However, this restriction is often alleviated by retaining a fairly small subset of the training dataset from the former tasks. The retained subset is denoted by the exemplar set in the literature.

The purpose of having an exemplar set for Class-IL is to address the issue of catastrophic forgetting [GMX+13, KPR+17, CDAT18]. The exemplars need to be properly selected to represent the previous tasks while managing limited resources and model training time. Beyond a simplistic random selection, several selection strategies were proposed in the literature, the most well-known being mean-of-features sampling, also called herding [RKSL17], entropy-based sampling [CDAT18],

and distance-based sampling [CDAT18].

In this chapter, we propose a novel sample selection strategy to maximize exemplars diversity for Class-IL. The proposed method is inspired from the coherence measure for sparse approximation introduced for adaptive kernel-based models. To the best of our knowledge, this is the first time that the coherence is investigated for DL, and more specifically for Class-IL. To this end, we define the coherence between two images as a normalized inner product between their feature vectors obtained by a deep neural network, *e.g.*, employing a CNN as a feature extractor. Besides providing a solution for selecting diverse and informative samples using the coherence measure, our sampling strategy takes into account a fixed memory budget for each class as a selection criterion. Therefore, the proposed method ensures that the selected exemplars are not only diverse but also accommodate memory limits, which is critical in the IL context. Several experiments are conducted to compare our method with the aforementioned existing sample selection techniques, on the two well-known datasets CIFAR-100 and MIT Indoor-67. The obtained results show that our method outperforms state of the art techniques, with better average test accuracy in low computational complexity.

The contributions of this chapter are summarized as follows:

- We introduce a novel sample selection strategy based on the coherence measure to maximize the exemplars diversity. The proposed algorithm is relevant for Class-IL, as it relies on a fixed memory budget for each class.

- This is the first time that the coherence measure is defined in a DL framework, beyond linear and kernel-based models. We bring forward the advantages of the coherence measure, initially explored in the literature of compressed sensing and sparse approximation, to Class-IL.

- We provide some theoretical results, allowing to connect the proposed coherence criterion to the herding criterion, and demonstrate the relevance of the proposed method with extensive comparative experiments.

- We shed light on the importance of Class-IL for maintaining and updating a vision-based indoor localization system.

The rest of this chapter is organized as follows. In Section 5.2, we explain the value of Class-IL for vision-based indoor localization systems. In Section 5.3, we

introduce the foundations of Class-IL in the context of image classification in addition to commonly used sample selection strategies. Section 5.4 exhibits the proposed coherence-based sample selection strategy. In Section 5.5, we evaluate the performance of our sample selection approach on two different datasets. Finally, Section 5.6 concludes this chapter.

5.2 Class-incremental Learning for Indoor Scene Recognition

Regular monitoring and maintenance of vision-based indoor localization systems is important for ensuring that they continue to provide accurate and reliable results. To maintain an effective and robust indoor scene recognition model, there are several important maintenance points that should be addressed, including:

- Data collection and annotation: To keep the system up-to-date, new indoor scene data collection and annotation need to be done regularly.
- Model(s) training: Retraining the model when needed can improve its performance and adapt it to changes in indoor environments.
- Model(s) evaluation: Regularly testing the model may help identify and address any performance issues.
- Data management: Efficiently managing large amounts of indoor scene data is crucial to ensure system's performance and scalability.

Retraining the indoor scene recognition model with new data when necessary can maintain the localization system's performance. When new rooms must be added to the system, Class-IL must be used. This enables the system to adapt to new classes without requiring the entire model to be retrained from scratch. In this context, Class-IL can help in several ways:

- Adaptation to changes in the indoor environment: Indoor environments can change as new objects and scenes may appear over time. Class-IL enables the system to adapt to these changes in an efficient manner, by incrementally adding new classes to the indoor scene recognition model, rather than retraining it from scratch.

- Maintained accuracy: The indoor localization system requires continuous knowledge update to maintain system accuracy and ensure that predictions are always up-to-date.
- Reduced computational cost: Retraining a DL model from scratch can be computationally expensive. Class-IL allows the system to adapt to new classes incrementally, reducing the computational cost and making the system more practical for real-world applications.
- Better scalability: As the system continues to learn, it may encounter a large number of new classes, making it challenging to store all of the information in memory. A sample selection strategy enables the system to manage its memory resources, as it only stores exemplars from seen classes. Furthermore, Class-IL reduces the demand for significant expansion of the model size, unlike ensemble multiple models techniques that increase memory requirements with the increase in the number of tasks [SAR19, RPR20].

Class-IL techniques were not explicitly built and tested for indoor scene recognition, but they can be adapted and applied to this problem domain to help with the maintenance of localization systems.

5.3 Background on Class-incremental Learning

In this section, we provide a comprehensive overview of Class-IL in the field of DL. We start by laying the essential foundations of Class-IL, investigating the knowledge distillation technique. Then, we delve into the existing sample selection strategies.

5.3.1 Foundations of Class-incremental Learning

Data is collected incrementally in real-life. Of particular interest is emerging new classes, namely when new data from previously-unknown classes need to be learned. Deep neural networks in general, and more specially CNNs, have a fixed supervised learning mechanism that cannot learn new classes. To address this issue, the learning process must be continuous and capable of taking in new inputs. However, due to the high processing needs and the unavailability of all previously seen training samples because of memory management, retraining CNNs can be challenging. Memory management constraints restrict the storage of all past training

samples, especially in scenarios with large datasets or limited memory resources. This restriction gets more important as incremental updates become more crucial.

To enable continuous learning and adaptability in CNNs, new methodologies and techniques are required [SAR19] instead of training a model from scratch for each update. In the context of image classification using CNNs, the concept of Class-IL is essential [MLT+22]. Class-IL techniques seek to effectively incorporate new data into CNN models by learning from new classes while retaining knowledge gained from earlier classes. Researchers are innovating Class-IL algorithms to enhance CNN performance when adapting to new classes and prevent catastrophic forgetting, *i.e.*, a phenomenon that occurs when previously gained knowledge is lost when training on new data [GMX+13, KPR+17, CDAT18]. The goal of these models is to be able to dynamically adapt to new information while maintaining high classification accuracy and successfully handling changing data distributions. Overall, Class-IL algorithms need to meet the following criteria [PUUH01]:

- The model should be able to learn additional information from new data and to accommodate to new classes that may be introduced.
- The model should not require access to the original data, used to train the existing classifier.
- The model should preserve previously acquired knowledge.

To address this issue, the rehearsal-based strategy has been advocated to maintain few samples from previously seen classes. For this purpose, a number of studies relying on response-based Knowledge Distillation (KD) [HVD15] have recently emerged, such as Learning without Forgetting (LwF) [LH17], incremental Classifier and Representation Learning (iCaRL) [RKSL17], Bias Correction (BiC) [WCW+19], and Deep Model Consolidation (DMC) [ZZG+20].

LwF [LH17] was proposed by Li and Hoiem to incrementally train a single network to learn multiple tasks (*i.e.,* multiple sets of classes) without using samples from old classes, preventing catastrophic forgetting by applying the distillation loss. This loss was introduced in [HVD15] for model compression in which knowledge is distilled from a large pretrained model (called teacher) to a smaller one (called student). In LwF, the student is trained on the new classes while also learning from the teacher's predictions on these classes, which assists the student in mimicking the behavior of the teacher, allowing the knowledge to be retained. When using LwF

distillation concept with exemplars (denoted by LwF-E), the distillation loss is applied to the selected exemplars from old classes as well as the data from new classes. This loss is then combined with the classification loss [MLT+22].

Following the notation of [RKSL17], let $\mathrm{X}^1, \mathrm{X}^2, \ldots$, be the sample sets where all images of a set $\mathrm{X}^y = \{x_1^y, \ldots, x_{n_y}^y\}$ are of class $y \in \mathbb{N}$, n_y being the number of samples from that class. Let P^y be the set of selected exemplars for class y, with the number of stored exemplars per class equal to a fixed parameter m. For a new task, new data $\mathrm{X}^s, \ldots, \mathrm{X}^t$ for new classes $s, \ldots, t$ (disjoint from previous or future classes) are encountered. We have to update the old model $\Theta^{1:s-1}$ to get a new one $\Theta^{1:t}$ able to classify old and new classes $1, \ldots, s-1, s, \ldots, t$. The new model is obtained by using a distillation loss and a classification loss, as described in the following.

Let D be the training set containing both the training dataset of the classes $s, \ldots, t$ (*i.e.*, new task) and the exemplars from the previous classes $1, \ldots, s-1$ (*i.e.*, old tasks), namely

$$\mathrm{D} = \bigcup_{y=s,\ldots,t} \{(x,y) : x \in \mathrm{X}^y\} \cup \bigcup_{y=1,\ldots,s-1} \{(x,y) : x \in \mathrm{P}^y\}. \tag{5.1}$$

Let $\hat{o}(x) = [\hat{o}_1(x), \ldots, \hat{o}_{s-1}(x)]$ be the output logits of the old model $\Theta^{1:s-1}$ trained on the old classes and $o(x) = [o_1(x), \ldots, o_s(x), \ldots, o_t(x)]$ the output logits of the new model $\Theta^{1:t}$. The distillation loss is applied on the combined training set D defined in (5.1) following [LH17]:

$$\mathrm{L}_d = -\sum_{x \in \mathrm{D}} \sum_{k=1}^{s-1} \hat{\pi}_k(x) \log[\pi_k(x)], \tag{5.2}$$

where $\hat{\pi}_k(x)$ and $\pi_k(x)$ are temperature scaled logits for class $k \in [1, \ldots, s-1]$, defined respectively as

$$\hat{\pi}_k(x) = \frac{e^{\frac{\hat{o}_k(x)}{\mathrm{T}}}}{\sum_{j=1}^{s-1} e^{\frac{\hat{o}_j(x)}{\mathrm{T}}}} \quad \text{and} \quad \pi_k(x) = \frac{e^{\frac{o_k(x)}{\mathrm{T}}}}{\sum_{j=1}^{s-1} e^{\frac{o_j(x)}{\mathrm{T}}}}, \tag{5.3}$$

with T the temperature scaling parameter.

The softmax cross-entropy is used as the classification loss, which is computed as follows:

$$\mathrm{L}_c = -\sum_{(x,y) \in \mathrm{D}} \sum_{k=1}^{t} \delta_{k=y} \log[p_k(x)], \tag{5.4}$$

where $\delta_{k=y}$ is the indicator function (*i.e.*, ground truth of the image) and $p_k(x)$ is the

output softmax probability defined by,

$$p_k(x) = \frac{e^{o_k(x)}}{\sum_{j=1}^{t} e^{o_j(x)}}. \tag{5.5}$$

The overall loss is formulated as follows:

$$\mathrm{L} = \mathrm{L}_c + \lambda \mathrm{L}_d, \tag{5.6}$$

where λ is the distilling coefficient used to balance between the two terms.

5.3.2 Sample Selection Strategies

Due to restricted memory capacity, it is not possible to store all data for network training replay. Thus, to manage memory limits, avoid forgetting and maintain knowledge from previous classes, the selection of representative samples becomes important. Although there are several Class-IL techniques proposed in the literature [BPK21], we focus primarily on the sample selection strategies adopted in Class-IL, since it is the main topic studied in this paper.

In Class-IL, rehearsal-based techniques require storing and repeating past data to prevent catastrophic forgetting. The sample selection procedure involves selecting a collection of representative samples from the classes that best represent the class distribution and features. These chosen exemplars are subsequently saved in the separate memory, where they serve as a reference knowledge for the old classes during IL. Selected exemplars are important as they affect the efficiency and effectiveness of IL algorithms. Care must be taken to ensure that the chosen exemplars effectively capture the key characteristics of their classes while minimizing the risk of catastrophic forgetting. Uncertainty and diversity are two different factors of sample selection that are often considered to select the most representative and informative exemplars.

The updated CNN model $\Theta^{1:t}$ is interpreted as a trainable feature extractor ϕ followed by a classification layer and a softmax layer with the same number of output nodes as the number of learned classes so far (in this case t nodes). Uncertainty-based selection strategies depend on the network's predictions (*i.e.,* softmax output probabilities), while diversity-based selection ones depend on the network's intermediate feature vectors (*i.e.,* embeddings at the output of the feature extrac-

tor). Beyond a simple random selection, researchers have proposed several sample selection strategies, the most well-known being the mean-of-features sampling, also known as herding [RKSL17], the entropy-based sampling [CDAT18], and the distance-based sampling [CDAT18].

Herding [Wel09, RKSL17] chooses exemplars repeatedly for each class by selecting those with features that are closest to the features mean. For each sample of a class, embeddings are extracted, and the mean of all feature vectors of this class samples is calculated by,

$$\mu_y = \frac{1}{n_y} \sum_{x \in \mathrm{X}^y} \phi(x). \tag{5.7}$$

At each iteration, an exemplar is chosen so that, when added to the other selected exemplars in its class, the resulting exemplars mean is the closest to μ_y. This is repeated for all the classes that we want to select exemplars for.

Entropy-based sampling [CDAT18] exploits the uncertainty information to select exemplars by using entropy of the model's predictions (*i.e.,* output probabilities). It computes the entropy of the softmax outputs and selects exemplars that have a higher uncertainty (*i.e.,* higher entropy) for each class. The uncertainty information of a sample is calculated based on,

$$\text{Entropy}(x) = -\sum_{k=1}^{t} p_k(x) \log [p_k(x)], \tag{5.8}$$

with $p_k(x)$ being the output softmax probability as defined in (5.5). Samples with high entropies are more difficult and confusing for the model and thus are considered more informative and potentially beneficial training data for future incremental steps.

Distance-based sampling [CDAT18] selects exemplars based on their distance to the model's decision boundary. This sample selection strategy prioritizes selecting samples that are closer to the decision boundary for each class. The smaller the distance measure, the less confidently the model predicts the class, and vice versa. For a given sample $x \in \mathrm{X}^y$, the distance from the decision boundary is calculated by,

$$d_y(x) = \phi(x)^\intercal \; w^y, \tag{5.9}$$

where w^y are the last FC layer parameters for class y. Samples close to the decision boundary are considered more challenging and thus are selected to help the model to learn better in the next incremental steps.

5.4 Coherence-based Criterion for Sample Selection

In this section, we delve into the integration of coherence measure in sample selection for Class-IL highlighting its importance in capturing the relationships between feature vectors. Then, we present our coherence-based sample selection approach with some theoretical insights.

5.4.1 Introduction to The Coherence Measure

The coherence measure is fundamental in many disciplines, including signal processing and machine learning [RSS23]. It is central in sparse models, providing theoretical foundations and practical insights, such as in compressed sensing [BCKV15]. With the emergence of online learning, where new samples become available in real-time, maintaining sparsity turns out to be challenging, requiring the selection of relevant samples for model formulation. These contributing samples, known as atoms in the literature, are typically organized in a set referred to as a dictionary. The construction of a meaningful dictionary and the measurement of its relevance have been explored in the literature with the coherence criterion.

The coherence measure corresponds to the largest correlation between atoms of a given dictionary. The coherence of a dictionary of m unit-norm atoms $x_1, x_2, \ldots, x_m$ is defined as

$$\text{coh} = \max_{i \neq j} |\langle x_i, x_j \rangle|. \tag{5.10}$$

This simple measure allows a deep analysis and characterization of the quality of the dictionary for sparse analysis and synthesis. In spite of introducing the coherence measure for linear sparse models [RSS23], it has been investigated for nonlinear (shallow) kernel-based models in several studies, providing in-depth theoretical results [Hon15a, Hon15b].

Motivated by the underlying theoretical results of the coherence measure, several online algorithms were proposed, such as nonlinear adaptive filtering [RBH09] and nonlinear principal component analysis [Hon12], to name a few. To derive online learning algorithms, one needs to select incrementally the atoms of the dictionary that has a coherence below a user-defined threshold $\gamma \in [0;1]$, namely,

$$\text{coh} \leq \gamma. \tag{5.11}$$

This condition enforces an upper bound on the inner product measure between

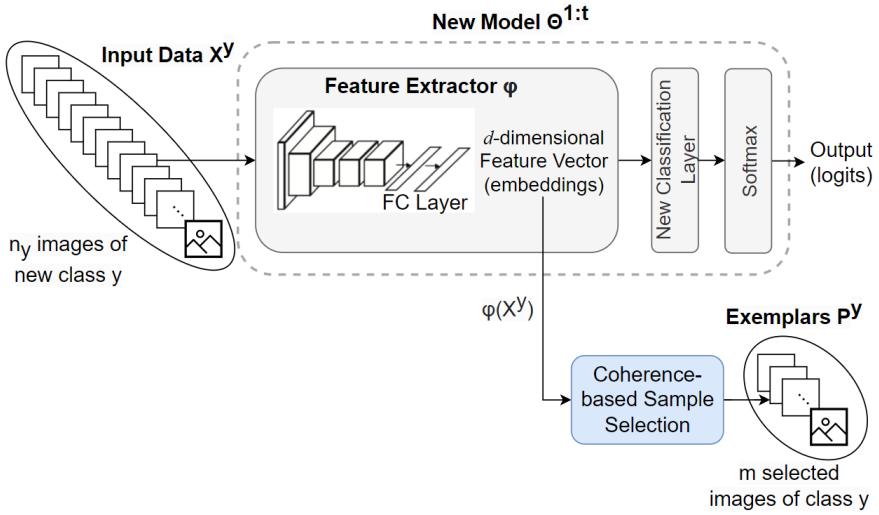


Figure 5.1: Coherence-based sample selection strategy.

each pair of atoms, or feature vectors when working in a feature space. The threshold γ controls the level of diversity of the selected samples, where a null value yields an orthogonal basis (*i.e.*, most dissimilar samples).

5.4.2 Proposed Coherence Measure in DL

The proposed coherence-based sample selection strategy is represented in Figure 5.1. The main building block is the investigation of the coherence measure within the DL embedding. By considering the coherence as a measure of similarity of DL feature vectors, we provide deeper insights, as opposed to the shallow versions of the coherence as given in (5.10) or its kernel-based counterpart. Let ϕ denote the feature extractor from the deep neural network, from some input space $\mathscr{X}$ to a feature space $\mathbb{R}^d$ of dimension d, namely $\phi : \mathscr{X} \rightarrow \mathbb{R}^d$. Then the coherence measure is defined as

$$\text{coh} = \max_{i \neq j} |\langle \phi(x_i), \phi(x_j) \rangle|, \tag{5.12}$$

if the embeddings are unit-norm; Otherwise, replace $\phi(x)$ with $\phi(x)/||\phi(x)||$. The higher the inner product between two embeddings of a pair of images, the more similar these images are.

To the best of our knowledge, the present paper is the first one that explores the coherence measure beyond shallow models (*i.e.*, linear and kernel-based models), by investigating a coherence criterion for Class-IL in a DL model. Therefore, in order to have a diverse set of exemplars, we aim to select the least mutually coherent

exemplars, as described in the following with the coherence criterion.

5.4.3 Proposed Coherence-based Sample Selection

We propose a novel sample selection method relying on the coherence criterion to boost the performance of Class-IL. The proposed coherence-based sample selection selects a diverse group of representative exemplars of a class by retaining samples that are mutually least coherent. As represented in Figure 5.1, this sampling technique is based on feature vectors obtained from the DL embedding, namely the last trained model so far, which means after the last incremental step. It strives to capture the sample diversity, guaranteeing that the chosen exemplars support effective information retention and learning in the IL context.

Rather than relying on a fixed threshold γ on the coherence between exemplars as given in (5.11), we propose to work with a fixed memory budget for each class. Indeed, memory management is critical in Class-IL to handle the limited resources available for storing exemplars while accommodating the addition of new classes, ensuring the model's ability to learn incrementally and retain knowledge from both old and new classes throughout incremental steps. With the proposed strategy, the memory allocation grows incrementally as new classes are available. Each class is assigned its own fixed memory space to store m exemplars. When a new class is added, a portion of the memory is allocated for storing exemplars of that specific class to avoid catastrophic forgetting. The specific memory requirements are determined by the IL scenario, including device memory limits, the number of classes, and the desired trade-off between memory usage and knowledge preservation. By having a fixed memory budget per class m, this strategy enables us to select representative diverse samples while respecting the memory budget.

For a new task consisting of a new set of classes $s,\dots,t$, an update procedure is called as data for these new classes is available. The procedure adjusts the DL parameters to get the new model $\Theta^{1:t}$ based on KD explained in Section 5.3.1 that will then be used to augment the exemplars saved in memory to get $\mathrm{P}^s,\dots,\mathrm{P}^t$ based on the new training data $\mathrm{X}^s,\dots,\mathrm{X}^t$. For each new class $y \in [s,\dots,t]$, we compute the inner product between each pair of vectors to investigate the relationships between these feature vectors (*i.e.,* class-wise comparisons). Let G be the Gram matrix containing the inner products $\langle \phi(x_i), \phi(x_j) \rangle$ for $x_i, x_j \in \mathrm{X}^y$. Since we have n_y vectors and we need the inner product for each pair, we only need to compute and store

Algorithm 2 Coherence-based sample selection

Input: image set $\mathrm{X}^y = \{x_1^y, \ldots, x_{n_y}^y\}$ of a class $y \in [s, \ldots, t]$, current feature extractor function $\phi : \mathscr{X} \to \mathbb{R}^d$.
for each: $y \in [s, \ldots, t]$
  Compute Gram matrix G for all entries $x_i^y, x_j^y \in \mathrm{X}^y$
  while size(G) $> m \times m$ **do**
    Select $(i, j) = \mathrm{argmax}_{i \neq j} |\mathrm{G}_{ij}|$
    Update $\mathrm{X}^y \leftarrow \mathrm{X}^y \setminus \{x_i^y\}$
    Remove i^{th} row and i^{th} column from G
  end while
Output: exemplar set $\mathrm{P}^y \leftarrow \mathrm{X}^y$.

half of the Gram matrix entities since it is symmetric.

The proposed sample selection algorithm is performed once for each of the classes $y \in [s, \ldots, t]$ as described in Algorithm 2 which guarantees the diversity of selected exemplars for each class. In order to produce a γ-coherent set of m exemplars for each class y, our method aims to eliminate the maximum coherence values. To this end, the Gram matrix G is first computed. From this matrix, the maximum coherence value is identified, pinpointing the pair of images that exhibits the strongest coherence. Then, the class's training set X^y is updated by eliminating the image indexed equivalently to the index of the maximum coherence value found. Simultaneously, the corresponding row and column are removed from G. This sequence of steps is repeated until reaching a Gram matrix of size $m \times m$, aligning with the predefined parameter m. Thus, for a given dataset X^y after an incremental step and a fixed budget per class equal to m, coherence-based sample selection selects exemplars for each new class to get at the end the set P^y to be used in addition to the previously stored exemplars in next incremental steps.

5.4.4 Theoretical Analysis

By eliminating the maximum coherence values, the proposed coherence-based criterion aims to select m exemplars per class with the least mutual coherence γ. In the following, we provide an upper bound on the approximation error of the mean-of-features μ_y of all the samples of the class defined in (5.7). This theoretical result provides connections with the herding criterion which aims to approximate μ_y [Wel09, RKSL17].

Theorem 5.4.1. *Consider the coherence-based criterion that selects m exemplars of coherence γ from the n_y samples of X^y. Let μ_{P^y} denote the approximation of μ_y by these exemplars. The error of this approximation is upper-bounded as follows*

$$\left\| \mu_y - \mu_{\mathrm{P}^y} \right\| \leq \left(1 - \frac{m}{n_y}\right) \sqrt{\max_{x \in \mathrm{X}^y} ||\phi(x)||^2 - \gamma}. \tag{5.13}$$

Proof. The proof follows the same reasoning as in the proof of Theorem 1 in [NHR12]. Let $\mathscr{P}$ be the projection operator onto the space spanned by the m exemplars, thus $\mu_{\mathrm{P}^y} = \mathscr{P}\mu$. Then, we have from the generalized triangular inequality

$$\begin{aligned}
\|\mu_y - \mu_{\mathrm{P}^y}\| &= \left\| \frac{1}{n^y} \sum_{x_i \in \mathrm{X}^y} (1 - \mathscr{P})\phi(x_i) \right\| \\
&\leq \sum_{x_i \in \mathrm{X}^y} \frac{1}{n^y} \|\phi(x_i) - \mathscr{P}\phi(x_i)\| \\
&= \frac{1}{n^y} \sum_{x_i \in \mathrm{X}^y \setminus \mathrm{P}^y} \|\phi(x_i) - \mathscr{P}\phi(x_i)\|
\end{aligned}$$

where the last equality is due to the fact that $\|\phi(x_i) - \mathscr{P}\phi(x_i)\| = 0$ for all $x_i \in \mathrm{P}^y$. Furthermore, the Pythagorian theorem allows to write

$$\|\phi(x_i) - \mathscr{P}\phi(x_i)\|^2 = \|\phi(x_i)\|^2 - \|\mathscr{P}\phi(x_i)\|^2.$$

The first term in the right-hand-side is upper-bounded by $\max_{x \in \mathrm{X}^y} ||\phi(x)||^2$. The second term, namely the square norm of the projection of $\phi(x)$, corresponds to the maximum inner product $\langle \phi(x), \varphi \rangle$ over all the unit-norm vectors φ, which leads to

$$\begin{aligned}
\|\mathscr{P}\phi(x_i)\|^2 &= \max_{\zeta} \frac{\sum_{x_j \in \mathrm{P}^y} \zeta_j \langle \phi(x_k), \phi(x_i) \rangle}{\| \sum_{x_j \in \mathrm{P}^y} \zeta_j \phi(x_j) \|} \\
&\geq \max_{x_k} \frac{|\langle \phi(x_k), \phi(x_i) \rangle|}{\langle \phi(x_k), \phi(x_k) \rangle},
\end{aligned}$$

where the inequality results from a specific distribution of the coefficients. From the coherence-based criterion, the right-hand-side is lower-bounded by γ, which concludes the proof.

□

5.5 Experiments and Results

In this section, we evaluate the performance of the proposed coherence-based sample selection strategy compared to other existing sample selection techniques typically employed for Class-IL.

5.5.1 Datasets

We study the performance of different sample selection strategies used in rehearsal-based Class-IL on two different datasets: CIFAR-100 [KH+09] and MIT Indoor-67 [QT09].

CIFAR-100 provides 32×32 color (RGB) images for 100 object classes, with 600 images divided into 500 for training and 100 for testing for each class. Following [MLT+22], a padding of 4 was added to each side of the image for data augmentation, and crops of 32×32 were randomly selected during training while center crops were adopted during testing. Additionally, input normalization and random horizontal flipping were performed. We randomly selected 50 classes from the CIFAR-100 dataset with 5 tasks of 10 classes each.

MIT Indoor-67 includes 67 indoor scene categories with 15,620 color (RGB) images in total (refer to Table 2.1 for more details about this dataset). The number of images varies across categories with at least 100 images per category divided into 80 images for training and 20 images for testing. Images were resized to 256×256 with random crops of 224×224 for training and center crops for testing then normalization was applied, as corroborated by [MLT+22]. This dataset has 5 tasks divided as follows: (0, 14) (*i.e.*, 14 classes for the initial task), (1, 14), (2, 13), (3, 13) and (4, 13).

5.5.2 Experimental Setup

We adopt LwF with exemplars (denoted by LwF-E) as a Class-IL technique. We implemented the distillation loss L_d following (5.2) based on [MLT+22] PyTorch source code [1] We fixed the temperature scaling parameter to $T = 2$ as proposed in [MLT+22] and in most of the literature. The distilling coefficient λ was fixed to 1. We relied on pretrained ResNet-32 [HZRS16] with CIFAR-100 dataset and pretrained MobileNet-v2 [SHZ+18] with MIT Indoor-67. The coherence expression given in (5.10) fur-

[1] Framework for Analysis of Class-Incremental Learning (FACIL); GitHub repository: `https://github.com/mmasana/FACIL`, accessed on 16 January 2023.

ther simplifies because the last layer of these deep neural networks imposes non-negativity with a ReLU operation. We used a learning rate search scheme [DLT21], a patience of 10, a learning rate factor of 3 (*i.e.,* the learning rate was divided by this factor each time the patience is exhausted), a gradient clipping at 10 000, a SGD optimizer with a momentum of 0.9, a weight decay of 0.0002 and a training batch size of 64 samples as in [MLT+22]. The training phase stopped either if the learning rate became equal to 10^{-4} or if the training reached 200 epochs.

5.5.3 Performance Evaluation

5.5.3.1 Accuracy

In order to analyze the overall Class-IL process and access performances at task p, the task agnostic average test accuracy metric is used [MLT+22]. The test accuracy is defined as,

$$\text{accuracy}_p = \frac{1}{p+1} \sum_{q=0}^{p} a_{p,q}, \tag{5.14}$$

where $a_{p,q}$ denotes the accuracy of task q after learning task p, with 0 being the initial task ($q \leq p$). Five runs were conducted to evaluate the performance.

We compare our proposed strategy in the fixed memory per class scenario taking $m = 20$ exemplars per class with existing and validated sample selection algorithms that have previously been employed in Class-IL approaches, mainly: herding [RKSL17], entropy [CDAT18] and distance [CDAT18]. Note that the comparison between the different sample selection strategies is based on identical data splits. The average test accuracies, denoted as accuracy_p ($_{avg}$(%)), are presented in Table 5.1 for the four sample selection strategies. Results show that the proposed coherence-based sample selection approach outperforms the reference strategies in terms of test accuracy which demonstrates the effectiveness of our method in a Class-IL scenario.

5.5.3.2 Execution Time

We also measured the execution time of the sample selection algorithms, recording the duration of the selection process. The experiments were conducted on Google Colab, utilizing the NVIDIA T4 GPU provided by the platform. We evaluated the time taken for a selection of 20 exemplars per class after the initial task, which means for 10 classes of CIFAR-100 and for 14 classes of MIT Indoor-67. The results represented

Table 5.1: Accuracy rates, averaged over 5 runs, for LwF-E using different sampling strategies (best results are in bold).

Task	sampling strategy	CIFAR-100	MIT Indoor-67
p = 2 (after 3 tasks)	herding	35.12 ± 2.54	68.92 ± 1.76
	entropy	27.76 ± 1.00	69.02 ± 2.35
	distance	27.14 ± 2.44	68.70 ± 1.49
	coherence (ours)	**35.98 ± 2.18**	**69.26 ± 2.57**
p = 4 (after 5 tasks)	herding	28.84 ± 1.46	58.18 ± 1.17
	entropy	19.96 ± 0.73	56.58 ± 1.67
	distance	19.28 ± 1.02	57.14 ± 1.03
	coherence (ours)	**29.82 ± 2.71**	**58.52 ± 0.91**

Table 5.2: Time (seconds) needed for selecting 20 exemplars per class after the initial task (*i.e.,* for 10 classes from CIFAR-100 and 14 classes from MIT-Indoor-67) using different sampling strategies (best results are in bold).

sampling strategy	CIFAR-100	MIT Indoor-67
herding	4.78	9.10
entropy	2.08	**8.52**
distance	**1.86**	9.08
coherence (ours)	3.10	9.30

in Table 5.2 revealed that our coherence-based sample selection strategy exhibited reasonable average time selection compared to the other sampling strategies.

5.6 Conclusion

In this chapter, we have investigated the efficiency of the sample selection process, which plays a critical role in improving model performance and memory management in Class-IL. We provided a novel sample selection technique based on coherence criterion for increasing diversity among class exemplars. The proposed strategy seeks to promote incoherence by maximizing the distinctiveness of the selected exemplars for each class. The results show that the proposed method not only demonstrated superior performance compared to state of the art sample selection techniques in terms of accuracy, but also showcased a reasonable execution time.

Regardless of the DL model being used, this methodology can be seamlessly incorporated into any neural network architecture to ensure the diversity of the selected exemplars based on the feature representations. Future work includes expanding the proposed sample selection strategy analysis on other benchmarks in

order to encompass a broader perspective on its applicability.

Chapter 6

Conclusions and Future Work

"Everything is theoretically impossible until it is done."

Robert A. Heinlein

Sommaire

6.1 General Conclusion

Indoor localization has evolved into an important aspect of our daily lives, providing a diverse range of systems. Our research efforts throughout this thesis have led us to critical analysis and decisions regarding technologies and algorithms. These decisions have effectively enabled us to propose a novel indoor localization approach, achieving the objectives fixed at the beginning of this thesis.

In **Chapter 2**, we have reviewed various aspects of indoor localization, starting with an extensive overview of state of the art methods in the field. We then shifted our focus towards vision-based indoor localization, delving deeper into the intricacies of indoor scene recognition. We conducted a thorough examination of the integration of DL, particularly CNNs, for vision-based indoor localization. This investigation encompassed an analysis of existing approaches, available public datasets, as well as a comprehensive exploration of the associated limitations and challenges.

Additionally, in **Chapter 3**, we explored the utilization of built-in smartphone sensors for indoor localization. Here, we provided an extensive evaluation of the capabilities and constraints inherent to these sensors when applied to localization tasks. Furthermore, we presented examples of systems that leverage smartphone cameras and magnetic field sensors for indoor localization. We demonstrated the interoperability of DL frameworks for smartphone deployment. We also conducted a critical review of the advantages and disadvantages associated with the prevailing computing approaches employed in this specific context.

Chapter 4 presented our main contributions on indoor scene recognition. We introduced a novel direction-driven multi-CNN system for indoor scene recognition, specifically tailored for room-level localization. We used embedded smartphone sensors to consider the user's smartphone's magnetic heading relative to the magnetic North. Our work included the creation of a dedicated dataset with images annotated with magnetic heading values, along with a comparison of two heading-based weighted fusion techniques. The experimental results clearly demonstrated the superiority of our proposed system over the baseline, which solely relied on images. Notably, when applied to indoor scene image data, our system outperformed traditional CNN-based image classification. It is important to note that our system leverages built-in smartphone sensors, whose quality and accuracy can vary across different devices and environments. We also investigated the impact of magnetic heading errors, highlighting the resilience and stability of our approach. Additionally, we delved into current computing paradigms, examining their relevance to DL tasks. We explored the effect of model partitioning on computational efficiency and introduced a hybrid computing strategy, involving partial offloading between the mobile device and a server.

In **Chapter 5**, we presented novel contributions to Class-IL. We focused on the efficiency of the sample selection process, a crucial aspect for enhancing model performance and memory management in Class-IL. We presented an innovative sample selection technique based on the coherence criterion, aiming to maximize diversity among class exemplars. This strategy promotes incoherence by emphasizing the distinctiveness of selected exemplars within each class. Our experimental results not only showcased the superior accuracy of the proposed method compared to state of the art sample selection techniques but also demonstrated reasonable execution times. Importantly, this methodology can be seamlessly integrated into various neural network architectures to ensure diverse exemplar selection based on

feature representations.

6.2 Perspectives

Many challenges and limitations remain in achieving a fully functional, high-performance indoor localization system. This thesis outlines numerous study paths that we believe will be required to address in future research. As discussed in [SHG+15], a significant portion of the code in numerous ML systems is not primarily focused on the training and prediction processes. The development and deployment of the models in applications involves a number of auxiliary tasks and infrastructure components that represent a significant portion of the implementation (see Figure 6.1). As part of short and mid-term perspectives, we would like to investigate the following aspects of improving our proposed solutions:

- **Implementing a complete Android application:** As we have seen throughout this thesis, all implementations and experiments were carried out on MATLAB and Google Colab (Python). Although a first version of the Android application has been developed and used for dataset acquisition, one of our goals is to develop the full smartphone application for more real-world tests and performance assessments. To that purpose, we will soon be working on the design of this application. We will focus on improving the performance of the proposed indoor localization system and optimizing it for real-time Android application (*i.e.,* to achieve real-time processing, the processing time must be less than the acquisition time between two images).

- **Testing the developed application by end-users:** We understand how critical it is to validate the application's usability, functionality, and performance. To accomplish this, we plan to perform extensive testing with end-users, in order to meet real-world requirements.

As for long-term perspectives, we look forward to:

- **Diversifying DL models in the proposed system:** Due to the rapid progress of DL and computer vision, it was not possible to thoroughly investigate every available method. Scene recognition is complex, owing mostly to its intrinsic variety, which makes labeling scenes a challenging task. To strengthen the

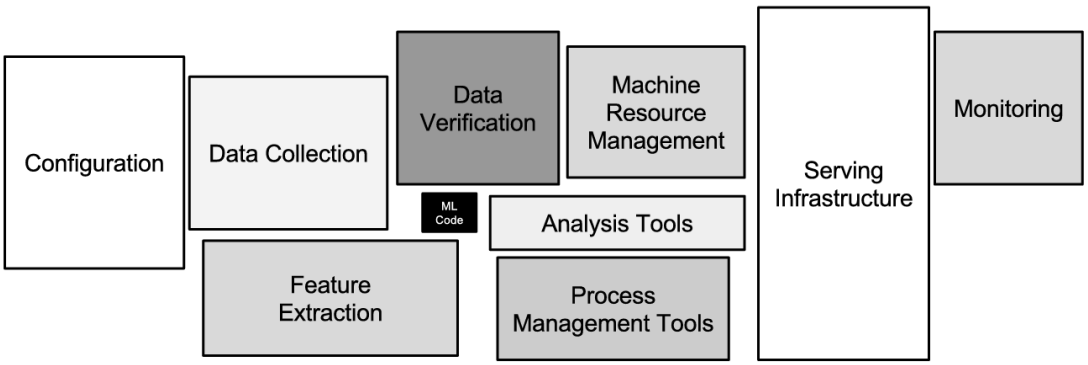


Figure 6.1: The required surrounding infrastructure for application implementation [SHG+15].

method proposed in Chapter 4, we intend to widen the frontiers of our indoor localization system by using varied computer vision DL models other than CNNs. This strategic expansion intends to investigate novel approaches and architectures that can improve our system's performance.

- **Integrating and fusing additional smartphone sensor data:** We hope to improve our localization system by including a broader range of relevant smartphone sensor data. Our goal is to fully utilize the capabilities of modern smartphones (*e.g.,* using barometer for floor identification). The enhanced sensor fusion technique will not only improve our system's accuracy, but will also allow it to adapt to dynamic indoor environment more efficiently. We hope that this extensive integration of sensor data will open up new possibilities beyond simple room-level localization, potentially covering context-aware services and enhanced user experiences in indoor spaces.

- **Extending the coherence-based sample selection evaluation:** We will extend the analysis of the sample selection strategy proposed in Chapter 5 to other benchmark datasets, providing a broader perspective on its applicability and potential improvements. We also aim to use the coherence-based sample selection for indoor localization, resulting in more accurate localization systems in changing indoor environments.

- **Exploring new DL architectures:** We plan to investigate cutting-edge DL approaches such as Transformers, which are currently gaining prominence. These architectures offer new possibilities for enhancing indoor localization techniques. Recent research articles in this domain reflect the ongoing evo-

lution of these methods, underscoring the significance of staying up-to-date with the latest advancements [GTP23, SAA+23]. In [GTP23], the authors proposed a novel framework for WiFi fingerprinting-based indoor localization based on vision transformer neural networks. The proposed method addresses the issue of the heterogeneity of wireless transceivers across various smartphones utilized by users. In [SAA+23], vision transform with dual multiscale attention is proposed to extract features at multiscales by processing large and small image patches.

Bibliography

[AASA20] Mouna Afif, Riadh Ayachi, Yahia Said, and Mohamed Atri. Deep learning based application for indoor scene recognition. *Neural Processing Letters*, 51:2827–2837, 2020. 37, 38, 39

[AAW+20] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020. 3, 15

[ACH18] Daniel AlShamaa, Farah Chehade, and Paul Honeine. Tracking of mobile sensors using belief functions in indoor wireless networks. *IEEE Sensors Journal*, 18(1):310–319, 2018. 5, 18, 55

[ACHC20] Daniel AlShamaa, Farah Chehade, Paul Honeine, and Aly Chkeir. An evidential framework for localization of sensors in indoor environments. *Sensors*, 20(1):318, 2020. 5, 18, 55

[AHP18] Imran Ashraf, Soojung Hur, and Yongwan Park. mpilot-magnetic field strength based pedestrian indoor localization. *Sensors*, 18(7):2283, 2018. 5, 15, 19

[AHP19] Imran Ashraf, Soojung Hur, and Yongwan Park. Application of deep convolutional neural networks and smartphone sensors for indoor localization. *Applied Sciences*, 9(11):2337, 2019. xii, 47, 51, 52

[AHP20] Imran Ashraf, Soojung Hur, and Yongwan Park. Smartphone sensor based indoor positioning: Current status, opportunities, and future challenges. *Electronics*, 9(6):891, 2020. 4, 5, 19, 45, 50, 51

[AKSM22] Khalid Abdulaziz Alnowibet, Imran Khan, Karam M Sallam, and Ali Wagdy Mohamed. An efficient algorithm for data parallelism based on stochastic optimization. *Alexandria Engineering Journal*, 61(12):12005–12017, 2022. 76

[AL20] Damir Arbula and Sandi Ljubic. Indoor localization based on infrared angle of arrival sensor network. *Sensors*, 20(21):6278, 2020. 5, 17

[AM22] Safar M Asaad and Halgurd S Maghdid. A comprehensive review of indoor/outdoor localization solutions in iot era: Research challenges and future perspectives. *Computer Networks*, 212:109041, 2022. 15

[AMCHC20] Daniel Alshamaa, Farah Mourad-Chehade, Paul Honeine, and Aly Chkeir. An evidential framework for localization of sensors in indoor environments. *Sensors*, 20(1):318, 2020. 53

[AZH+21] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021. 8

[BCKV15] Holger Boche, Robert Calderbank, Gitta Kutyniok, and Jan Vybíral. *A Survey of Compressed Sensing*, pages 1–39. Springer International Publishing, Cham, 2015. 96

[BETVG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 6, 27, 49

[Ble97] Geoffrey Blewitt. Basics of the gps technique: observation equations. *Geodetic applications of GPS*, 1:46, 1997. 4

[BPK21] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021. 94

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 6, 27, 47

[BSD+21] Wadii Boulila, Mokhtar Sellami, Maha Driss, Mohammed Al-Sarem, Mahmood Safaei, and Fuad A Ghaleb. Rs-dcnn: A novel distributed convolutional-neural-networks based-approach for big remote-sensing image classification. *Computers and Electronics in Agriculture*, 182:106014, 2021. 76

[CBLC20] Lei Chen, Kanghu Bo, Feifei Lee, and Qiu Chen. Advanced feature fusion algorithm based on multiple convolutional neural network for scene recognition. *CMES-Computer Modeling in Engineering & Sciences*, 122(2), 2020. 38

[CDAT18] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018. 88, 89, 92, 95, 102

[Chi13] C Chien. *The Hall effect and its applications*. Springer Science & Business Media, 2013. 52

[CMGCCL+11] Enrique Costa-Montenegro, Francisco J González-Castaño, David Conde-Lagoa, Ana Belén Barragáns-Martínez, Pedro S Rodríguez-Hernández, and Felipe Gil-Castiñeira. Qr-maps: An efficient tool for indoor user location based on qr-codes and google maps. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 928–932. IEEE, 2011. 47

[CPC16] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. 28

[CR19] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019. 60, 61, 62, 67, 77

[CVPA22] Domenica Costantino, Gabriele Vozza, Massimiliano Pepe, and Vincenzo Saverio Alfio. Smartphone lidar technologies for surveying and reality modelling in urban scenarios: Evaluation methods, performance and challenges. *Applied System Innovation*, 5(4):63, 2022. 44

[CWG+20] Shuyan Cheng, Shujun Wang, Wenbai Guan, He Xu, and Peng Li. 3dlra: An rfid 3d indoor localization method based on deep learning. *Sensors*, 20(9):2731, 2020. 5, 17

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 24, 30, 31, 37, 38, 72, 80

[DDVPR13] Pasquale Daponte, L De Vito, F Picariello, and M Riccio. State of the art and future developments of measurement applications on smartphones. *Measurement*, 46(9):3291–3307, 2013. xi, 43

[DH72] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. 49

[DLT21] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8250–8259, 2021. 102

[DLY+17] Xintao Ding, Yonglong Luo, Qingying Yu, Qingde Li, Yongqiang Cheng, Robert Munnoch, Dongfei Xue, and Guorong Cai. Indoor object recognition using pre-trained convolutional neural network. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–6. IEEE, 2017. 21

[DPD23] Vincenzo Di Pietra and Paolo Dabove. Recent advances for uwb ranging from android smartphone. In *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 1226–1233. IEEE, 2023. 44

[DPHB21] Andrea Daou, Jean-baptiste Pothin, Paul Honeine, and Abdelaziz Bensrhair. Amélioration des performances des réseaux de neurones convolutifs en localisation indoor par augmentation des données. In *ORASIS 2021*, 2021. 39

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005. 27

[EKRS13] Pablo Espinace, Thomas Kollar, Nicholas Roy, and Alvaro Soto. Indoor scene recognition by a mobile robot through adaptive object detection. *Robotics and Autonomous Systems*, 61(9):932–947, 2013. 26

[EVGW+10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 30

[FFCM17] André Filipe Gonçalves Gonçalves Ferreira, Duarte Manuel Azevedo Fernandes, Andre Paulo Catarino, and Joao L Monteiro. Localization and positioning systems for emergency responders: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2836–2870, 2017. 15

[FHS18] M Aideed Fadzilla, Azizi Harun, and AB Shahriman. Localization assessment for asset tracking deployment by comparing an indoor localization system with a possible outdoor localization system. In *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, pages 1–6. IEEE, 2018. 3, 15

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 29

[GCY+19] Guangyi Guo, Ruizhi Chen, Feng Ye, Xuesheng Peng, Zuoya Liu, and Yuanjin Pan. Indoor smartphone localization: A hybrid wifi rtt-rss ranging approach. *IEEE Access*, 7:176767–176781, 2019. xii, 53, 54

[GFM21] Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, pages 4416–4425. International Joint Conferences on Artificial Intelligence, 2021. 16

[GGL+16] Cole Gleason, Anhong Guo, Gierad Laput, Kris Kitani, and Jeffrey P Bigham. Vizmap: Accessible visual information through crowdsourced map reconstruction. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 273–274, 2016. xii, 47, 48

[GJVD18] Andrei Dmitri Gavrilov, Alex Jordache, Maya Vasdani, and Jack Deng. Preventing model overfitting and underfitting in convolutional neural networks. *International Journal of Software Science and Computational Intelligence (IJSSCI)*, 10(4):19–28, 2018. 66

[GLGL18] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7:87–93, 2018. 21

[GMX+13] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 88, 92

[GPMSSA21] Pedro J García-Paterna, Alejandro S Martínez-Sala, and Juan Carlos Sánchez-Aarnoutse. Empirical study of a room-level localization system based on bluetooth low energy beacons. *Sensors*, 21(11):3665, 2021. 8

[GTP23] Danish Gufran, Saideep Tiku, and Sudeep Pasricha. Vital: Vision transformer neural networks for accurate smartphone heterogeneity resilient indoor localization. *arXiv preprint arXiv:2302.09443*, 2023. 109

[GWCZ18] Wei Guo, Ran Wu, Yanhua Chen, and Xinyan Zhu. Deep learning scene recognition method based on localization enhancement. *Sensors*, 18(10):3376, 2018. 47, 66

[HBF19] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK*, pages 191–202. Springer, 2019. 31

[HDO+98] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 36

[HJLT23] Jiahao Huang, Steffen Junginger, Hui Liu, and Kerstin Thurow. Indoor positioning systems of mobile robots: A review. *Robotics*, 12(2):47, 2023. 14

[HK10] Daniel Hauschildt and Nicolaj Kirchhof. Advances in thermal infrared localization: Challenges and solutions. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–8. IEEE, 2010. 5, 17

[HKBA16] Munawar Hayat, Salman H Khan, Mohammed Bennamoun, and Senjian An. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, 25(10):4829–4841, 2016. 36

[HKLK17] Hyunwoo Hwangbo, Jonghyuk Kim, Zoonky Lee, and Soyean Kim. Store layout optimization using indoor positioning system. *International Journal of Distributed Sensor Networks*, 13(2):1550147717692585, 2017. 14

[HKPH23] Alexander Heinrich, Sören Krollmann, Florentin Putz, and Matthias Hollick. Smartphones with uwb: Evaluating the accuracy and reliability of uwb ranging. *arXiv preprint arXiv:2303.11220*, 2023. 44

[HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Pro-*

ceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017. 28

[HNO13] Michael Hölzl, Roland Neumeier, and Gerald Ostermayer. Analysis of compass sensor accuracy on several mobile devices in an industrial environment. In *International Conference on Computer Aided Systems Theory*, pages 381–389. Springer, 2013. 82, 83, 84

[Hon12] Paul Honeine. Online kernel principal component analysis: a reduced-order model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1814 – 1826, September 2012. 96

[Hon15a] Paul Honeine. Analyzing sparse dictionaries for online learning with kernels. *IEEE Transactions on Signal Processing*, 63(23):6343–6353, 2015. 96

[Hon15b] Paul Honeine. Approximation errors of online sparsification criteria. *IEEE Transactions on Signal Processing*, 63(17):4700–4709, 2015. 96

[HQY+22] Yuming Hu, Feng Qian, Zhimeng Yin, Zhenhua Li, Zhe Ji, Yeqiang Han, Qiang Xu, and Wei Jiang. Experience: Practical indoor localization for malls. In *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking*, pages 82–93, 2022. 14

[HSC+19] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 33

[Hu15] F Hu. Emerging techniques in vision-based indoor localization. *The City University of New York*, 2015. 20

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 92

[HWC22] Chen He, Ruiping Wang, and Xilin Chen. Rethinking class orders and transferability in class incremental learning. *Pattern Recognition Letters*, 161:67–73, 2022. 88

[HWT+20] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020. 33, 35

[HZC+17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. xi, 33, 34, 35, 72

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. xi, 28, 29, 101

[IHM+16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. xi, 33, 72, 81

[JD17] Jichao Jiao and Zhongliang Deng. Deep combining of local phase quantization and histogram of oriented gradients for indoor positioning based on smartphone camera. *International Journal of Distributed Sensor Networks*, 13(1):1550147716686978, 2017. 47

[JJCS15] Ruoxi Jia, Ming Jin, Zilong Chen, and Costas J Spanos. Soundloc: Accurate room-level indoor localization using acoustic signatures. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 186–193. IEEE, 2015. 8

[JQF+20] Ping Ji, Danyang Qin, Pan Feng, Tingting Lan, and Guanyu Sun. Research on indoor scene classification mechanism based on multiple descriptors fusion. *Mobile Information Systems*, 2020:1–14, 2020. 35

[JS23] Jehn-Ruey Jiang and Hanas Subakti. An indoor location-based augmented reality framework. *Sensors*, 23(3):1370, 2023. 15

[JSZ+15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 31, 32

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 30, 101

[KH14] Wonho Kang and Youngnam Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal*, 15(5):2906–2916, 2014. 5, 19

[KHB+16] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Roberto Togneri, and Ferdous A Sohel. A discriminative representation of convolutional features for indoor scene recognition. *IEEE Transactions on Image Processing*, 25(7):3372–3383, 2016. 36, 38

[KHG+17] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News*, 45(1):615–629, 2017. 77

[KHP17] Salman H Khan, Munawar Hayat, and Fatih Porikli. Scene categorization with spectral features. In *Proceedings of the IEEE international conference on computer vision*, pages 5638–5648, 2017. 36

[KKAMAA19] Jayakanth Kunhoth, AbdelGhani Karkar, Somaya Al-Maadeed, and Asma Al-Attiyah. Comparative analysis of computer-vision and ble technology based indoor navigation systems for people with visual impairments. *International Journal of Health Geographics*, 18:1–18, 2019. 26

[KKAMAA20] Jayakanth Kunhoth, AbdelGhani Karkar, Somaya Al-Maadeed, and Abdulla Al-Ali. Indoor positioning and wayfinding systems: a survey. *Human-centric Computing and Information Sciences*, 10(1):1–41, 2020. 3, 4, 14, 15

[KNZC18] Jian Kuang, Xiaoji Niu, Peng Zhang, and Xingeng Chen. Indoor positioning based on pedestrian dead reckoning and magnetic field matching for smartphones. *Sensors*, 18(12):4142, 2018. 5, 19

[KPC15] Soufien Kammoun, Jean-Baptiste Pothin, and Jean-Christophe Cousin. An efficient fuzzy logic step detection algorithm for unconstrained smartphones. In *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 2110–2114. IEEE, 2015. 5

[KPHB18] Siloère Konlambigue, Jean-Baptiste Pothin, Paul Honeine, and Abdelaziz Bensrhair. Fast and accurate gaussian pyramid construction by extended box filtering. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 400–404. IEEE, 2018. 2, 6

[KPHB19] Silvère Konlambigue, Jean-Baptiste Pothin, Paul Honeine, and Abdelaziz Bensrhair. Performance evaluation of state-of-the-art filtering criteria applied to SIFT features. In *19th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Ajman, United Arab Emirates, 2019. 2, 6, 35

[KPR+17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 88, 92

[KRM+21] Viachaslau Kachurka, Bastien Rault, Fernando I Ireta Muñoz, David Roussel, Fabien Bonardi, Jean-Yves Didier, Hicham Hadj-Abdelkader, Samia Bouchafa, Pierre Alliez, and Maxime Robin. Weco-slam: Wearable cooperative slam system for real-time indoor localization under challenging conditions. *IEEE Sensors Journal*, 22(6):5122–5132, 2021. 19

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 24, 28, 31, 33, 36, 38, 47, 48, 72, 80

[KSV16] Shiu Kumar, Ronesh Sharma, and Edwin Vans. Localization for wireless sensor networks: A neural network approach. *arXiv preprint arXiv:1610.04494*, 2016. 55

[LBD+89] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 28

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 6, 28

[LCEVBGM20] Alejandro López-Cifuentes, Marcos Escudero-Vinolo, Jesús Bescós, and Álvaro García-Martín. Semantic-aware scene recognition. *Pattern Recognition*, 102:107256, 2020. 37

[LCL+20] Ming Li, Ruizhi Chen, Xuan Liao, Bingxuan Guo, Weilong Zhang, and Ge Guo. A precise indoor visual positioning approach using a built image feature database and single user image from smartphone cameras. *Remote Sensing*, 12(5):869, 2020. xii, 48, 51

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011. 6

[LCY13] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 48

[LCZ+22] Qing Li, Rui Cao, Jiasong Zhu, Xianxu Hou, Jun Liu, Sen Jia, Qingquan Li, and Guoping Qiu. Improving synthetic 3d model-aided indoor image localization via domain adaptation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 183:66–78, 2022. 26

[LDBL07] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007. 54

[LH17] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 92, 93

[LHRX20] Matthias Langer, Zhen He, Wenny Rahayu, and Yanbo Xue. Distributed training of deep learning models: A taxonomic perspective. *IEEE Transactions on Parallel and Distributed Systems*, 31(12):2802–2818, 2020. 76

[Lin12] Tony Lindeberg. Scale invariant feature transform. 2012. 6, 27

[Liu19] Dongqing Liu. *Mobile Data and Computation Offloading in Mobile Cloud Computing*. PhD thesis, Université de Technologie de Troyes; Université de Montréal, 2019. 60

[LLLJ14] Di Lin, Cewu Lu, Renjie Liao, and Jiaya Jia. Learning important spatial pooling regions for scene classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3726–3733, 2014. 35

[LLY+20] Fen Liu, Jing Liu, Yuqing Yin, Wenhan Wang, Donghai Hu, Pengpeng Chen, and Qiang Niu. Survey on wifi-based indoor positioning techniques. *IET communications*, 14(9):1372–1383, 2020. 5, 17, 18

[LMNF09] Vincent Lepetit, Francesc Moreno-Noguer, and P Fua. Epnp: Efficient perspective-n-point camera pose estimation. *Int. J. Comput. Vis*, 81(2):155–166, 2009. 49

[LRH+19] Ang Li, Xiaogang Ruan, Jing Huang, Xiaoqing Zhu, and Fei Wang. Review of vision-based simultaneous localization and mapping. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 117–123. IEEE, 2019. 19

[LSP06] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006. 24, 25, 35

[LSY+22] Wenjie Luo, Qun Song, Zhenyu Yan, Rui Tan, and Guosheng Lin. Indoor smartphone slam with learned echoic location features. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 489–503, 2022. 47

[LT19] Shaopeng Liu and Guohui Tian. An indoor scene classification method for service robot based on CNN feature. *Journal of Robotics*, 2019, 2019. 21

[LTC21] Raymond Low, Zeynep Duygu Tekler, and Lynette Cheah. An end-to-end point of interest (poi) conflation framework. *ISPRS International Journal of Geo-Information*, 10(11):779, 2021. 15

[LTP17] Christopher Langlois, Saideep Tiku, and Sudeep Pasricha. Indoor localization with smartphones: Harnessing the sensor suite in your pocket. *IEEE Consumer Electronics Magazine*, 6(4):70–80, 2017. xi, 45

[LW18] Qilong Li and Xiaohong Wang. Image classification based on sift and svm. In *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pages 762–765. IEEE, 2018. 6, 27

[LZL+16] Zhenguang Liu, Luming Zhang, Qi Liu, Yifang Yin, Li Cheng, and Roger Zimmermann. Fusion of magnetic and visual sensors for indoor localization: Infrastructure-free and more effective. *IEEE Transactions on Multimedia*, 19(4):874–888, 2016. 26, 51

[LZV+20] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*, 2020. 76

[MAG+02] Luis Moreno, Jose M Armingol, Santiago Garrido, Arturo De La Escalera, and Miguel A Salichs. A genetic algorithm for mobile robot localization using ultrasonic sensors. *Journal of Intelligent and Robotic Systems*, 34:135–154, 2002. 5

[MB14] Andreas C Müller and Sven Behnke. Learning depth-sensitive conditional random fields for semantic segmentation of rgb-d images. In *2014 IEEE International conference on robotics and automation (ICRA)*, pages 6232–6237. IEEE, 2014. 36

[Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 24

[MLT+22] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022. 92, 93, 101, 102

[MMFW14] Chadly Marouane, Marco Maier, Sebastian Feld, and Martin Werner. Visual positioning systems—an extension to movips. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 95–104. IEEE, 2014. 47

[MMH+21] MG Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey. *ACM Computing Surveys (CSUR)*, 54(8):1–37, 2021. 77

[MMM+20] Anca Morar, Alin Moldoveanu, Irina Mocanu, Florica Moldoveanu, Ion Emilian Radoi, Victor Asavei, Alexandru Gradinaru, and Alex Butean. A comprehensive survey of indoor localization methods based on computer vision. *Sensors*, 20(9):2641, 2020. 14, 26

[MMN22] Kiran Maharana, Surajit Mondal, and Bhushankumar Nemade. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1):91–99, 2022. 29

[MNP+16] Yongshik Moon, Soonhyun Noh, Daedong Park, Chen Luo, Anshumali Shrivastava, Seongsoo Hong, and Krishna Palem. Capsule: A camera-based positioning system using learning. In *2016 29th IEEE International System-on-Chip Conference (SOCC)*, pages 235–240. IEEE, 2016. 47

[MPI20] Massimo Merenda, Carlo Porcaro, and Demetrio Iero. Edge machine learning for ai-enabled iot devices: A review. *Sensors*, 20(9):2533, 2020. 61, 62

[MPS14] Luca Mainetti, Luigi Patrono, and Ilaria Sergi. A survey on indoor positioning systems. In *2014 22nd international conference on software, telecommunications and computer networks (SoftCOM)*, pages 111–120. IEEE, 2014. 14

[MSG+20] Raul Montoliu, Emilio Sansano, Arturo Gascó, Oscar Belmonte, and Antonio Caballer. Indoor positioning for monitoring older adults at home: Wi-fi and ble technologies in real scenarios. *Electronics*, 9(5):728, 2020. 5, 17

[MWY+22] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022. 21

[MXD+18] Wei Ma, Hanjiang Xiong, Xuefeng Dai, Xianwei Zheng, and Yan Zhou. An indoor scene recognition-based 3d registration mechanism for real-time ar-gis visualization in mobile applications. *ISPRS international journal of geo-information*, 7(3):112, 2018. 21

[MZM+21] Bo Miao, Liguang Zhou, Ajmal Saeed Mian, Tin Lun Lam, and Yangsheng Xu. Object-to-scene: Learning to transfer object knowledge to indoor scene recognition. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2069–2075. IEEE, 2021. 37

[MZZS18] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 33

[NHR12] Zineb Noumir, Paul Honeine, and Cédric Richard. One-class machines based on the coherence criterion. In *Proc. IEEE workshop on Statistical Signal Processing (SSP)*, pages 600 – 603, Ann Arbor, Michigan, USA, 5 - 8 August 2012. 100

[NJKN20] Saad Naeem, Noreen Jamil, Habib Ullah Khan, and Shah Nazir. Complexity of deep convolutional neural networks in mobile computing. *Complexity*, 2020:1–8, 2020. 8

[NKP18] Muzammal Naseer, Salman Khan, and Fatih Porikli. Indoor scene understanding in 2.5/3d for autonomous agents: A survey. *IEEE access*, 7:1859–1887, 2018. xi, 20, 21

[NP17] Lucie Novakova and Terry L Pavlis. Assessment of the precision of smart phones and tablets for measurement of planar orientations: A case study. *Journal of Structural Geology*, 97:93–103, 2017. 83, 84

[OAM22] Guanglie Ouyang and Karim Abed-Meraim. Analysis of magnetic field measurements for indoor positioning. *Sensors*, 22(11):4014, 2022. 5, 19, 51

[PACK22] Giovanni Pau, Fabio Arena, Mario Collotta, and Xiangjie Kong. A practical approach based on bluetooth low energy and neural networks for indoor localization and targeted devices' identification by smartphones. *Entertainment Computing*, 43:100512, 2022. xii, 55, 56

[PDXL21] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021. 28

[PEH19] Alwin Poulose, Odongo Steven Eyobu, and Dong Seog Han. An indoor position-estimation algorithm using smartphone imu sensor data. *Ieee Access*, 7:11165–11177, 2019. 19

[Pet09] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009. 6, 27

[PMMRSP13] José Antonio Puértolas Montañés, Adriana Mendoza Rodríguez, and Iván Sanz Prieto. Smart indoor positioning/location and navigation: A lightweight approach. 2013. 47

[PUUH01] Robi Polikar, Lalita Upda, Satish S Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural

networks. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, 31(4):497–508, 2001. 92

[QL17] Jun Qi and Guo-Ping Liu. A robust high-accuracy ultrasound indoor positioning system based on a wireless sensor network. *Sensors*, 17(11):2554, 2017. 5

[QRLB20] Facundo Quiroga, Franco Ronchetti, Laura Lanzarini, and Aurelio F Bariviera. Revisiting data augmentation for rotational invariance in convolutional neural networks. In *Modelling and Simulation in Management Sciences: Proceedings of the International Conference on Modelling and Simulation in Management Sciences (MS-18)*, pages 127–141. Springer, 2020. 32

[QT09] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009. 25, 101

[R+01] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001. 47

[RBH09] Cédric Richard, José C. M. Bermudez, and Paul Honeine. Online prediction of time series data with kernels. *IEEE Transactions on Signal Processing*, 57(3):1058 – 1067, March 2009. 96

[RF21] Miguel Rêgo and Pedro Fonseca. Occ based indoor positioning system using a smartphone camera. In *2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 31–36. IEEE, 2021. 47

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 48

[RK22] Alvin Riady and Gede Putra Kusuma. Indoor positioning system using hybrid method of fingerprinting and pedestrian dead reck-

oning. *Journal of King Saud University-Computer and Information Sciences*, 34(9):7101–7110, 2022. 19

[RKSL17] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 88, 92, 93, 95, 99, 102

[RP00] Maximilian Riesenhuber and Tomaso Poggio. Models of object recognition. *Nature neuroscience*, 3(11):1199–1204, 2000. 21

[RPR20] Deboleena Roy, Priyadarshini Panda, and Kaushik Roy. Tree-cnn: a hierarchical deep convolutional neural network for incremental learning. *Neural Networks*, 121:148–160, 2020. 91

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 6

[RSS23] David Ramírez, Ignacio Santamaría, and Louis Scharf. *Coherence: In Signal Processing and Machine Learning*. Springer Nature, 2023. 96

[RW17] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017. 30

[SAA+23] Yahia Said, Mohamed Atri, Marwan Ali Albahar, Ahmed Ben Atitallah, and Yazan Ahmad Alsariera. Scene recognition for visually-impaired people's navigation assistance based on vision transformer with dual multiscale attention. *Mathematics*, 11(5):1127, 2023. 109

[SAR19] Syed Shakib Sarwar, Aayush Ankit, and Kaushik Roy. Incremental learning in deep convolutional neural networks using partial network sharing. *IEEE Access*, 8:4615–4628, 2019. 91, 92

[SD19] G Sreenu and MA Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1):1–27, 2019. 21

[SFH+19] Xudong Song, Xiaochen Fan, Xiangjian He, Chaocan Xiang, Qianwen Ye, Xiang Huang, Gengfa Fang, Liming Luke Chen, Jing Qin, and Zumin Wang. Cnnloc: Deep-learning based indoor localization with wifi fingerprinting. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 589–595. IEEE, 2019. 5, 17

[SGD13] Kalyan Pathapati Subbu, Brandon Gozick, and Ram Dantu. Locateme: Magnetic-fields-based indoor localization using smartphones. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):1–27, 2013. 19, 50

[SHC+11] Georg Schroth, Robert Huitl, David Chen, Mohammad Abu-Alqumsan, Anas Al-Nuaimi, and Eckehard Steinbach. Mobile visual location recognition. *IEEE Signal Processing Magazine*, 28(4):77–89, 2011. 46

[Shc14] Maxim Shchekotov. Indoor localization method based on wi-fi trilateration technique. In *Proceeding of the 16th conference of fruct association*, pages 177–179, 2014. 18

[SHC+19] Hongje Seong, Junhyuk Hyun, Hyunbae Chang, Suhyeon Lee, Suhan Woo, and Euntai Kim. Scene recognition via object-to-scene class conversion: end-to-end training. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2019. 36

[SHG+15] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28, 2015. xiii, 107, 108

[SHK20] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Fosnet: An end-to-end trainable deep neural network for scene recognition. *IEEE Access*, 8:82066–82077, 2020. 37, 38

[SHK22] Jakob Schyga, Johannes Hinckeldeyn, and Jochen Kreutzfeldt. Meaningful test and evaluation of indoor localization systems in semi-controlled environments. *Sensors*, 22(7):2797, 2022. 16

[SHZ+18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. xi, 33, 34, 35, 72, 81, 101

[SK19] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 32

[SLA+18] Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600*, 2018. 76

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 28

[SLK18] Kwangjae Sung, Dong Kyu 'Roy' Lee, and Hwangnam Kim. Indoor pedestrian localization using ibeacon and improved kalman filter. *Sensors*, 18(6):1722, 2018. 19

[SM16] Guowei Shi and Ying Ming. Survey of indoor positioning systems based on ultra-wideband (uwb) technology. In *Wireless Communications, Networking and Applications: Proceedings of WCNA 2014*, pages 1269–1278. Springer, 2016. 5, 17

[SP20] Petros Spachos and Konstantinos N Plataniotis. Ble beacons for indoor positioning at an interactive iot-based smart museum. *IEEE Systems Journal*, 14(3):3483–3493, 2020. 55

[SP22] Claudio Filipi Gonçalves Dos Santos and João Paulo Papa. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Computing Surveys (CSUR)*, 54(10s):1–25, 2022. 31, 66

[SS18] Sebastian Sadowski and Petros Spachos. Rssi-based indoor localization with the internet of things. *IEEE access*, 6:30149–30161, 2018. 18

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 28

[TAC+21] Nestor Michael Tiglao, Melchizedek Alipio, Roselia Dela Cruz, Fawaz Bokhari, Sammia Rauf, and Saad Ahmad Khan. Smartphone-based indoor localization techniques: State-of-the-art and classification. *Measurement*, 179:109349, 2021. 5, 19, 51

[TC22] Zeynep Duygu Tekler and Adrian Chong. Occupancy prediction using deep learning approaches across multiple space types: A minimum sensing strategy. *Building and Environment*, 226:109689, 2022. 17

[Tho16] Martin Thoma. A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541*, 2016. 21

[TKT+22] Dimitrios Tsourounis, Dimitris Kastaniotis, Christos Theoharatos, Andreas Kazantzidis, and George Economou. Sift-cnn: When convolutional neural networks meet dense sift descriptors for image and sequence classification. *Journal of Imaging*, 8(10):256, 2022. 27

[TKVT18] Thomas Tegou, Ilias Kalamaras, Konstantinos Votis, and Dimitrios Tzovaras. A low-cost room-level indoor localization system with easy setup for medical applications. In *2018 11th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 1–7. IEEE, 2018. 8

[TL19] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 28, 37

[TLG+20] Zeynep Duygu Tekler, Raymond Low, Burak Gunay, Rune Korsholm Andersen, and Lucienne Blessing. A scalable bluetooth low energy approach to identify occupancy patterns and profiles in office spaces. *Building and Environment*, 171:106681, 2020. 5, 17

[TMHF00] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer, 2000. 49

[TMU17] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Scale-invariant recognition by weight-shared cnns in parallel. In *Asian Conference on Machine Learning*, pages 295–310. PMLR, 2017. 31, 32

[TSYW17] Zhehang Tong, Dianxi Shi, Bingzheng Yan, and Jing Wei. A review of indoor-outdoor scene classification. In *2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017)*, pages 469–474. Atlantis Press, 2017. 20

[TWK17] Pengjie Tang, Hanli Wang, and Sam Kwong. G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition. *Neurocomputing*, 225:188–197, 2017. 36

[VS07] Julia Vogel and Bernt Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007. 21

[WA19] Yujie Wei and Burcu Akinci. A vision and learning-based indoor localization and semantic mapping framework for facility operations and management. *Automation in Construction*, 107:102915, 2019. 21

[WCW+19] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 374–382, 2019. 92

[WDZ+19] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10734–10742, 2019. 33, 35

[Wel09] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128, 2009. 95, 99

[WHA14] Shaohua Wan, Changbo Hu, and Jake K Aggarwal. Indoor scene recognition from rgb-d images by learning scene bases. In *2014 22nd International Conference on Pattern Recognition*, pages 3416–3421. IEEE, 2014. 37

[WHL+20] Xiaofei Wang, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2):869–904, 2020. 61, 62, 77

[WHS16] Martin Werner, Carsten Hahn, and Lorenz Schauer. Deepmovips: Visual indoor positioning using transfer learning. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, 2016. 47

[WKM11] Martin Werner, Moritz Kessel, and Chadly Marouane. Indoor positioning using smartphone camera. In *2011 international conference on indoor positioning and indoor navigation*, pages 1–6. IEEE, 2011. 47

[WLL18] Robert J Wang, Xiang Li, and Charles X Ling. Pelee: A real-time object detection system on mobile devices. *Advances in neural information processing systems*, 31, 2018. 33, 34

[XCD19] Song Xu, Wusheng Chou, and Hongyi Dong. A robust indoor localization system integrating visual localization aided by cnn-based image retrieval with monte carlo localization. *Sensors*, 19(2):249, 2019. 26

[XCL+18] Aoran Xiao, Ruizhi Chen, Deren Li, Yujin Chen, and Dewen Wu. An indoor positioning system based on static objects in large indoor scenes by using smartphone cameras. *Sensors*, 18(7):2229, 2018. xii, 48, 50

[XHE+10] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 25

[XLHL20] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 28

[XLL+18] Lin Xie, Feifei Lee, Li Liu, Zhong Yin, Yan Yan, Weidong Wang, Junjie Zhao, and Qiu Chen. Improved spatial pyramid matching for scene recognition. *Pattern Recognition*, 82:118–129, 2018. 36

[XLL+19] Mengwei Xu, Jiawei Liu, Yuanqiang Liu, Felix Xiaozhu Lin, Yunxin Liu, and Xuanzhe Liu. A first look at deep learning apps on smartphones. In *The World Wide Web Conference*, pages 2125–2136, 2019. 57

[XXZ+14] Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. Scale-invariant convolutional neural networks. *arXiv preprint arXiv:1411.6369*, 2014. 31, 32

[XZC+19] Chunwei Xia, Jiacheng Zhao, Huimin Cui, Xiaobing Feng, and Jingling Xue. Dnntune: Automatic benchmarking dnn models for mobile-cloud computing. *ACM Transactions on Architecture and Code Optimization (TACO)*, 16(4):1–26, 2019. 77

[XZYN16] Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M Ni. A survey on wireless indoor localization from the device perspective. *ACM Computing Surveys (CSUR)*, 49(2):1–31, 2016. 53

[YJHN07] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206, 2007. 35

[YN10] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010. 27

[YWLW20] Sifan Yang, Yue Wang, Yang Li, and Guijin Wang. Image enhancement and translation for rgb-d indoor scene recognition. In *Proceedings of the 2020 4th International Conference on Digital Signal Processing*, pages 7–11, 2020. 37

[YWW20] Xiao-min Yu, Hui-qiang Wang, and Jin-qiu Wu. A method of fingerprint indoor localization based on received signal strength difference by using compressive sensing. *EURASIP Journal on Wireless Communications and Networking*, 2020:1–13, 2020. 18

[ZBK+17] Yinda Zhang, Mingru Bai, Pushmeet Kohli, Shahram Izadi, and Jianxiong Xiao. Deepcontext: Context-encoding neural pathways for 3d holistic scene understanding. In *Proceedings of the IEEE international conference on computer vision*, pages 1192–1201, 2017. 39

[ZCS+23] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. 21

[ZDM+16] Fan Zhang, Fabio Duarte, Ruixian Ma, Dimitrios Milioris, Hui Lin, and Carlo Ratti. Indoor space recognition using deep convolutional neural network: a case study at mit campus. *arXiv preprint arXiv:1610.02414*, 2016. xii, 26, 47, 49

[ZGL19] Faheem Zafari, Athanasios Gkelias, and Kin K Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599, 2019. 5, 14

[ZHSS19] Zhenyong Zhang, Shibo He, Yuanchao Shu, and Zhiguo Shi. A self-evolving wifi-based indoor navigation system using smartphones. *IEEE Transactions on Mobile Computing*, 19(8):1760–1774, 2019. xii, 53, 54

[ZLK+17] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 25, 36, 37

[ZLT14] Dong Zhang, Dah-Jye Lee, and Brandon Taylor. Seeing eye phone: a smart phone-based indoor localization and guidance system for the visually impaired. *Machine vision and applications*, 25:811–822, 2014. 47

[ZLT+21] Delu Zeng, Minyu Liao, Mohammad Tavakolian, Yulan Guo, Bolei Zhou, Dewen Hu, Matti Pietikäinen, and Li Liu. Deep learning for scene classification: A survey. *arXiv preprint arXiv:2101.10531*, 2021. 36, 38

[ZLX+14] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27, 2014. 25

[ZLXQ18] Jinbo Zuo, Shuo Liu, Hao Xia, and Yanyou Qiao. Multi-phase fingerprint map based on interpolation for indoor localization using ibeacons. *IEEE sensors journal*, 18(8):3351–3359, 2018. 56

[ZWL16] Hongyuan Zhu, Jean-Baptiste Weibel, and Shijian Lu. Discriminative multi-modal feature fusion for rgbd indoor scene recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2969–2976, 2016. 37

[ZWQ+23] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 2023. 88

[ZWWN15] Yongpan Zou, Guanhua Wang, Kaishun Wu, and Lionel M Ni. Smartscanner: Know more in walls with your smartphone! *IEEE Transactions on Mobile Computing*, 15(11):2865–2877, 2015. 19, 50

[ZYT17] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2017. 6, 28

[ZZG+20] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140, 2020. 92

[ZZLS18] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. xi, 33, 34, 72, 81

[ZZXW19] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019. 21