Deep Learning to Assist Visually Impaired Individuals with Visual Exploration

Roger Girgis

Department of Electrical & Computer Engineering
McGill University
Montréal, Québec, Canada

April 2019

A thesis submitted to McGill University in partial fulfilment of the requirements for the degree of Master of Engineering.

2019/04/15

ABSTRACT

In navigating and understanding an outdoor environment, our world often requires the ability to see. People with visual impairments are, therefore, faced with significant challenges in exploring these environments. Deep learning has the potential to alleviate part of the frustrations they face. In this thesis, we assess the effectiveness of using deep learning to assist people with visual impairments.

One of the challenges faced by this user group is the crossing of intersections while remaining within the crosswalk. Veering-avoidance systems are an important assistive technology for visually impaired users, helping them when crossing intersections. The reliance of previous systems on specific features, such as zebra patterns or visible painted lanes, may be a significant factor behind their limited adoption. In this thesis, we design a mobile application that assists people with visual impairments in the task of crossing intersections. The application employs recent advances in machine learning, specifically deep learning, in combination with imitation learning. The use of convolutional neural networks makes our approach relatively independent of specific features. We start with the collection of demonstrative videos of intersection crossings executed by sighted individuals who simulated the process of veering. The collection is performed following a series of observational studies that allowed us to understand how blind people currently cross intersections. The individual video frames are labeled with the optimal crossing direction to gather the experts' recommended actions. A policy derivation technique is applied to extract the optimal behavior, resulting in an agent capable of providing the optimal crossing direction. Building on this agent, a prototype smartphone application is designed to provide users with real-time feedback both before and during intersection crossing. Results from our user study, conducted with eight blind participants, indicate that using the application significantly increases the probability that users will correctly align themselves before crossing, and reduces the probability that a user will veer outside the crosswalk. A series of iterative experiments were conducted with a blind individual to address the limitations we discovered through the user study. Our final solution employs a combination of inertial measurement unit sensors and the imitation learning agent.

Another challenge faced by visually impaired people is the understanding of the visual content in their immediate surrounding. Recent deep learning models offer methods that provide natural language descriptions of images. However, through our experience with such models, we found that they often provide irrelevant or inaccurate descriptions. The question of inaccurate descriptions relies on algorithmic improvements and better assessment strategies of the generated captions. To understand the relevancy problems, we review previous literature that provided descriptions of visual content using crowd-workers or automatic approaches. Through our review, we identify three main problems that need to be addressed in order to provide relevant descriptions: the information contained in the descriptions, the incorporation of crowd-workers to ensure accuracy of descriptions, and to provide users with efficient mechanisms for capturing images. We provide a vision of a mobile application that can result in the generation of an image description dataset geared towards assisting the blind community.

RÉSUMÉ SCIENTIFIQUE

La vision joue un rôle important dans notre capacité à se déplacer et comprendre le monde qui nous entoure. Les personnes malvoyantes font face à des défis importants lorsqu'il s'agit d'explorer l'environnement. L'apprentissage profond a le potentiel de réduire une partie des frustrations auxquelles elles sont confrontées. Dans cette thèse, nous évaluons le potentiel de l'apprentissage profond pour aider les personnes malvoyantes.

L'un des défis de ce groupe d'utilisateurs est le franchissement d'intersection tout en traversant sur le passage piéton. Les systèmes d'évitement d'obstacle sont une technique d'assistance importante pour les utilisateurs malvoyants lorsqu'ils traversent des intersections. Ces systèmes dépendent de variables spécifiques, telles que les motifs de zèbre ou les marquages au sol ce qui explique qu'ils sont peu utilisés. Dans cette thèse, nous concevons une application mobile qui aide les personnes malvoyantes à traverser les intersections. Cette application utilise les dernières avancées en matière d'apprentissage automatique, et en particulier d'apprentissage profond, en combinaison avec l'apprentissage par imitation. L'utilisation de réseaux de neurones à convolution rend notre approche relativement indépendante de certaines variables. Dans un premier temps nous avons collecté des vidéos de personnes voyantes traversant des intersections en simulant le processus d'évitement. La collecte est ralisée à la suite d'une série d'études d'observation qui nous a permis de comprendre comment les personnes aveugles traversent actuellement les intersections. Les sections (images) de vidéos individuelles sont étiquetées avec la direction optimale à prendre pour traverser l'intersection afin de rassembler les actions recommandées par les experts. Une technique de politique dérivative est appliquée pour extraire le comportement optimal, ce qui donne un agent capable de fournir la direction optimale pour traverser. Un prototype d'application pour smartphone est conçu pour fournir aux utilisateurs des informations en temps réel, avant et pendant le franchissement d'une intersection. Les résultats de notre étude, menée auprès de huit participants aveugles, indiquent que l'utilisation de l'application augmente considérablement la probabilité que les utilisateurs s'alignent correctement avant de traverser et réduisent la probabilité qu'un utilisateur quitte le passage piéton. Une série d'expériences itératives a été menée avec un individu aveugle afin de remédier aux limitations découvertes lors de l'étude. Notre solution finale est constituée de la combinaison de capteur de mesure d'inertie et de l'agent d'apprentissage par imitation.

Les personnes malvoyantes doivent également comprendre le contenu visuel de leur environnement immédiat. Les modèles récents d'apprentissage profond proposent des méthodes qui fournissent des descriptions d'images en langage naturel. Cependant, nous avons constaté qu'en pratique, ces modèles fournissaient souvent des descriptions non pertinentes ou inexactes. Cela pourrait être réglée avec l'amélioration des algorithmes utilisés pour résoudre ce genre de tâche mais aussi par l'utilisation de meilleures stratégies d'évaluation des sous-titres générés. Afin de comprendre la raison sous-jacente à ces limitations, nous avons réalisé une revue de littérature des méthodes utilisant des annotateurs humains ou des approches automatiques pour la génération de sous-titres à partir de contenu visuel. Cela nous a permis d'identifier trois problèmes principaux qui doivent être résolus afin de fournir des descriptions pertinentes: les informations contenues dans les descriptions, l'incorporation d'annotateurs humain pour s'assurer de l'obtention de descriptions précises et fournir aux utilisateurs des mécanismes efficaces pour la capture d'image. Finalement, nous décrivons une application mobile permettant la création d'un jeu de données de description d'images destiné à aider la communauté des aveugles.

ACKNOWLEDGEMENTS

I would like to thank all those who have nurtured my creativity throughout my life, and gave me the freedom to explore various paths throughout my academic career. A special thanks to my supervisor, Jeremy R. Cooperstock, for his research guidance and the freedom he provided me during my exploration of a research path.

I would also like to thank numerous present and former members of the Shared Reality Lab for pilot testing my walking straight application and for participating in numerous fruitful discussions throughout this research. A special thanks to Jeffrey R. Blum, Jan Anlauff, Antoine Weill-Duflos and Manfred Diaz in this regard. I would also like to thank Michael Ciarcello for being an active tester of the street crossing application, and for helping me understand the various challenges blind people face in exploring outdoor environments.

In addition, I would like to thank the Institut Nazareth et Louis-Braille for allowing us to study the problems in collaboration with them, and for helping with the recruitment of blind individuals in testing our application.

I would also like to acknowledge the funding from the Canadian Internet Registry Assocation's Community Investment Program, which helped support this research.

Contents

List of Figures

List of Tables

Preface

Section 2.1 of Chapter 2 was co-authored with Manfred Diaz, where we contributed to the research and writing equally.

Most of Chapter 4 was also co-authored with Manfred Diaz. Specifically, Sections 4.1 and 4.2 were published in the International Conference on Computer Vision's Workshop on Assistive Computer Vision and Robotics [10], where Manfred Diaz and I contributed equally to all parts of this work. Sections 4.3 and 4.4 are part of an unsuccessful submission to the conference on Human Factors in Computing Systems 2018 (CHI 2018), where Manfred Diaz contributed to parts of the design of the mobile application, and I focused on the experimental design with blind participants and preparation of the model for mobile deployment. We both had equal roles in writing of those sections. The experiments and the writing of Section 4.5 as well as the writing of Section 4.6 were performed by myself.

All other chapters were written by me.

Chapter 1

Introduction

Independent navigation and exploration of a city has long been a daunting task for visually impaired individuals. Guide dogs and canes have been the primary assistive devices used by the blind community to assist them with outdoor navigation. Though they are indispensable to the population, they cannot help the user with understanding new environments and navigating through them. This challenge is exacerbated when the environment the user is navigating is unknown. For this reason, they tend to remain in known environments [11], as they learn specific landmarks of those routes, such as intersection characteristics. Various assistive devices have been proposed to help people with visual impairments with the various tasks involved in this challenge, such as:

- *Navigation through outdoor environments*, e.g., talking GPS systems [12].
- *Safe crossing at intersections*, e.g., systems that employ inertial measurement sensors to provide feedback regarding heading during crossing [13].
- *Situational awareness*, e.g., systems that provide information about points of interest around the user [14].

While many research and development projects have been proposed to assist blind individuals with these challenges, they are not often used by the blind community [15]. There are a few factors that result in the lack of adoption of these solutions. One contributing factor could be the elevated costs of implementing the proposed solutions widely, e.g., accessible pedestrian signals for crossing intersections. We can also attribute it to the difficulty of the task at hand, e.g., providing accurate and relevant descriptions of the world around the user in situ. A recurring problem with the solutions being proposed is that they have not been properly tested with and tailored to the visually impaired community, i.e., ensuring adequate human-computer interaction.

The recent surge of deep learning research presents potential solutions to many of the problems faced by the community. In particular, convolutional neural network (CNN) architectures have been shown to outperform all other methods for image recognition and classification tasks [16, 17, 18, 19, 1]. They have also been shown to produce state-of-the-art results in object detection datasets [20, 21]. In addition, CNN architectures combined with recurrent neural networks (RNN) have been employed to produce accurate captions of images [2, 22, 23]. Indeed, these models have recently started being introduced into the assistive technology domain, such as in mobile applications [24, 25, 26], or employed in products developed by assistive technology companies for the blind [27, 28].

The ability to incorporate these models into mobile applications allows users the access to (relatively) reliable and cost effective solutions to the challenges they face. Other recent research in answering visual questions from visually impaired users [29, 30] provides a glimpse of the human-computer interaction problems that need attention. In this thesis, we apply and evaluate the suitability of deep learning on two of the tasks presented above: safe crossing at intersections and situational awareness.

1.1 Safe Crossing at Intersections

Crossing intersections is considered to be the most difficult and risky aspect of independent travel for visually impaired individuals [31]. Challenges include determining whether the intersection is one- or two-way, orienting in the correct direction for crossing, obtaining the status of the pedestrian signal, and knowing when veering is occurring while crossing. In this thesis, we focus on the last of these challenges.

The problem with visual impairments is that, in the absence of environmental cues, humans tend to walk in circles [32], often with diameters of less than 20 meters. This has obvious implications to crossing at intersections, which can be of similar length. Mobility training focuses on techniques to keep the individual walking as straight as possible while maintaining a safe distance from parallel traffic, i.e., remaining within the marked lines designating pedestrian crossings. This is accomplished to a certain extent by using the sound of traffic, as we discuss in Chapter 4. However, even after training, detection of veering remains difficult [33]. Guth et al. [34] assessed the skill level of experienced blind pedestrians in aligning themselves using traffic sounds, and found that they are useful but cannot guarantee accurate alignment. In addition, they noted a trial-to-trial variability large enough that every subject would have eventually walked out into the center of an intersection following a sufficient number of re-starts. Technologies developed to overcome these problems include accessible pedestrian signal (APS), embedded sensors, and mobile vision systems. As discussed in further detail in Chapter 2, the shortcomings of these previous efforts to address the veering problem for blind pedestrians motivated our deep learning vision-based approach described in this thesis.

The process of development can be described at a high-level in the following five phases:

1. We conduct observational sessions with visually impaired people to analyze the steps they follow when crossing at intersections.

2. We propose the use of four types of CNN models combined with imitation learning to learn the optimal policy based on sighted individuals' expert recommendation.

3. We implement an Android mobile application that employs auditory feedback to provide users with directional cues.

4. We conduct an experiment with blind individuals to evaluate the proposed appli-

cation. In analyzing the experimental data, we gained insights on requirements for future developments.

5. Using the lessons learned from (4), we present an iterative improvement process of the application where we periodically met with a visually impaired individual who tested the application and provided us with further insights for improvements.

1.2 Situational Awareness

A memorable scene in “Le fabuleux destin d’Amélie Poulain” is when Amélie assists a blind man with getting to the metro, while describing the surrounding environment. “We just passed the drum major’s widow! She’s worn his coat since the day he died…The horse’s head on the butcher’s wall has lost an ear …In the bakery window, there are lollipops!…Sugarplum ice cream at this shop…We’re passing the pork butcher, ham for 79 francs, Spareribs for 45…Now we arrive at the cheese shop …At the butcher’s, a baby is watching a dog that’s watching the chickens roasting…Now we’re in front of the newspaper kiosk by the metro.” The blind man was of course astounded, stunned and surprised by all the things he heard and learned as he walked down the street. This scene provides us with inspiration for what we should strive to obtain from automatic scene description models.

Exploration of a city inherently demands an understanding of the world around. As mentioned previously, guide dogs and white canes have long been the main assistive technology used by the community in outdoor navigation, but these have little impact on helping understand the surrounding environment. Many mobile systems, such as *Autour* [35] and *BlindSquare* [36], have been proposed to provide the user with information regarding points of interest. The problem with these systems is their reliance on GPS, which can exhibit errors with means of 10-30 meters in areas with surrounding buildings, as reported by Blum et al. [35, 37]. Google AI have also acknowledged the problem, noting in a recent article [38] that “[GPS in dense urban environments] can result in highly inaccurate placements on the map, meaning that your location could appear on the wrong side of the street, or even a few blocks away”. They also indicate that they are working on a solution they call “global localization, which combines Visual Positioning Service (VPS), [Google] Street View, and machine learning to more accurately identify position and orientation”.

A more appropriate approach to understanding the immediate surrounding would be to directly analyze the rich visual content of those scenes. In recent years, we have seen the introduction of many applications, whether mobile or desktop, that utilize images to provide different kinds of descriptions to blind users [39, 29, 5, 40]. For instance, both *Be My Eyes* [39] and *VizWiz* [29] allow the users to connect to remote sighted volunteers, who respond to a question regarding the visual content sent by the user. Through our experience with blind users while developing our iOS application *Autour* [41, 35], we have found that users desire the ability to obtain descriptions of the visual world around them. One user commented: “A dream would be if, in the future, we will be able to interact with the AR-technology and even let the app discover and recognize obstacles, colors, shapes and faces.” Another user mentioned that it is desirable to have access to a paid human description service that connects a user to a remote audio description service where a sighted visual describer awaits live feed from the user and provides descriptions of the environment, similar to audio description of movies and TV-shows, “like a personal Amélie Poulain.” It is clear that the blind population have a strong desire to understand the world through its visual content especially since our environment often requires the ability to see.

The surge of deep learning in recent years has resulted in the availability of many models capable of classifying images and generating descriptions of their contents. Unfortunately, this has led to the misperception that these models are ready to be incorporated into mobile applications and the problems of situational awareness for blind individuals would be solved. However, through our testing with different models that produce image descriptions, we have found that they often provide generic, irrelevant and/or erroneous descriptions of the contents in the scene. This is the case even with models that achieved state-of-the-art results in the image captioning task (such as the Neural Image Captioner [2] as of 2015).

There are multiple factors contributing to the poor performance of these models in practice. One such factor is that they were optimized using the maximum likelihood objective, which tends to generate the most likely, and therefore generic, solution [42, 43]. Addressing this phenomenon, both Shetty et al. [43] and Dai et al. [42] derive a generative adversarial network (GAN) that has the capability of generating diverse captions. However, the problem of irrelevant or erroneous descriptions cannot be circumvented by their proposed improvements. The questions of relevancy and accuracy of descriptions cannot be automatically discovered without having an oracle that knows the optimal image description for blind people, which of course does not exist.

These issues are especially problematic for people with visual impairments when used in outdoor environments as they can lead to a great deal of confusion to the user, who has minimal reference to the ground truth. As mentioned by one of our users, a potential use case of such models would be to provide information of roadwork and sidewalk closures, giving the user information of the surrounding that would allow them to choose the safest walking path. Unfortunately, these performance issues would exacerbate the frustrations of the blind community as we have no means of guaranteeing the relevancy or correctness of the descriptions provided by these models. We attribute part of the problem of relevancy to the fact that the training data has limited overlap with images acquired in the wild. Most available scene captioning models were trained on the Microsoft COCO dataset [44], a dataset not specifically designed for the visually impaired community. Recently, *VizWiz* [29] released a dataset [45] on visual question answering, collected through their application where blind people each took an image and recorded a spoken question about it, together with 10 crowd-sourced answers per visual question. The release of this dataset sparked a new computer vision challenge in the AI community with a workshop held at the European Conference on Computer Vision (ECCV) in 2018 [46] devoted to algorithmically learning this dataset. As of the time of writing this thesis, we are not aware of an equivalent image description dataset or challenge geared towards the blind community.

In the second part of this thesis, we study the current status of describing visual content to visually impaired people. The most common method to assess captioning models in the computer vision literature is using automatic assessment metrics such as BLEU [47] or using sighted human evaluators who rate the accuracy of the caption based on the image. We present these methods in Chapter 2 and discuss why they are not sufficient when the goal is to provide descriptions to blind people. In Chapter 5, we review previous research that tackled the problem of describing visual content specifically to people with visual impairments. We proceed to summarize the findings of those methods by providing the lessons learned, with a focus on the type of visual content that should be described, the framing of those descriptions and the human-computer interaction requirement for capturing visual content from a blind user's perspective. Finally, based on the lessons learned, we propose an outline of a mobile application that provides descriptions of visual content. Inspired by *VizWiz* [29] and *Be My Eyes* [39], the application would leverage sighted crowd workers to ensure the accuracy of the description, and utilizes blind users to ensure the relevancy and utility of those descriptions.

1.3 Author's Contribution

In this work, we explore the effectiveness of machine learning in various assistive technology domains aimed at aiding people living with visual impairments in navigating outdoor environments. Specifically, we focus on two challenges deemed important in outdoor exploration: (1) The use of deep imitation learning in crossing intersections while maintaining a path that keeps the user in the safe zone of the crosswalk; (2) Evaluating the current status of computer-generated descriptions of outdoor scenes against ones generated by sighted individuals. Concretely, the contributions of this work are as follows:

1. We propose the first mobile application that assists visually impaired individuals in crossing intersections with minimal veering using deep imitation learning. We demonstrate that the application can effectively assist this user population, especially when it is combined with built-in Inertial Measurement Unit (IMU) sensor data, in comparison to only using their mobility training.

2. Through a conducted literature review of prior work in scene understanding for visually impaired people, we identify the most important features that should be contained in a scene description as well as the optimal evaluation questions that should be posed to sighted and blind participants. To our knowledge, this is the first literature review investigating the question of optimal scene description content in order for blind people to find deep learning description systems useful.

Chapter 2

Literature Review

In this chapter, we review previously proposed assistive systems and on-going projects designed for people with visual impairments. We start by focusing on prior work that helps blind individuals with the task of crossing intersections, where we focus on systems that rely on IMU sensors to provide heading measurements as well as systems that employ computer vision. Next, we review current automatic assessment metrics and sighted human assessment methods of the quality of image captioning models.

2.1 Intersection Crossing and Veering-Avoidance Systems

Previous technologies have been developed to address the challenges associated with crossing intersections as a blind pedestrian. One of the best solutions available to date is the accessible pedestrian signal (APS). APS systems indicate *when* it is safe to cross [48], and in certain cases, offer auditory cues that help the user determine orientation. Guidelines suggest that the volume of such audible signals must be between 2 and 5 dB above the ambient noise conditions at a distance within 3.5 m from the sound source in order to avoid masking by ambient noise [49]. In practice, through our experience of working with visually impaired people, this criterion is not always respected. More problematically, due to their high cost estimated at over $25k (USD) per new installation, and approximately $8k (USD) at intersections with existing poles [50], APS deployment remains limited. For example, according to the Montreal Association for the Blind, only 133 APS systems have been installed in the city of Montreal, Canada, with 1875 intersections [51, 52].

Other systems have been developed in an attempt to tackle the veering and intersection-crossing problem encountered by the visually impaired community using relatively more widely deployable approaches. We categorize these systems into non-vision-based and vision-based technologies. While some are commercially available, many remain limited to the research setting, and are still either in the experimental phase or would be too expensive for widespread commercial deployment.

2.1.1 Non-vision-Based Systems

One approach to tackle this challenge is to employ embedded sensors, such as accelerometers or gyroscopes found in typical smartphones [13, 53, 33] to provide feedback with the goal of preventing veering. However, these systems may suffer from problems of sensor instability in addition to the potential need for frequent re-calibration, and do not address the challenge of initial alignment in the correct direction at the start of crossing.

Ross et al. [53] developed a wayfinding guidance system comprised of a computer, carried by the user in a backpack, with an array of speakers placed against the back, a digital compass mounted either on the shoulder or in a hat, and a pair of ear buds mounted on the hat. The array of speakers is used to provide the user with haptic feedback, while the ear buds provide the user with two types of auditory feedback. The authors compared three forms of directional feedback: 1) a spatialized tone rendered through the ear buds at repeated intervals; 2) spoken description of the direction, e.g., “one o’clock”; 3) a “shoulder tapping” method that provided directional haptic feedback through the activation of different speakers. A second variable from their study was the position of the compass, which was either placed on the shoulder to provide a body-referenced output or in a hat to provide a head-referenced output. The authors found that the shoulder-tapping method yielded the best performance, with the spatialized tone method coming as a close second. The spoken description of the direction produced the worst results, obtaining the least number of points across participants. In fact, the authors note that there was no significant difference in performance between the two. In addition, subjects differed in which mode produced the least veering during crossing. When participants used their best method, they achieved a 31% improvement in veering performance when compared to a no-feedback baseline. Another important finding from their work was that the shoulder tapping approach performed best when the sensor data driving the feedback came from a digital compass placed on the shoulder, rather than the user’s head. However, in the spatialized tone method, positioning the compass on the head allowed the users to perform slightly better than placing it on the shoulder. In both cases, the subjects preferred having the compass positioned on the shoulder.

Ross et al. [53] had initially attempted to augment the orientation signal by installing a special system at test intersections that would communicate with a detector installed in the backpack computer. However, this could not be accomplished due to county versus city jurisdiction over traffic lights issues. This further demonstrates the deployment difficulties of such technologies that require changes to city infrastructure. These practical difficulties would need to be overcome before the system could be widely deployed.

Guth [33] proposed the *Anti-Veering Training Device* (AVTD) that employs a solid-state gyroscope to measure the user’s cumulative rotation as they walk along a path. The gyroscope also provides tilt and temperature compensation for additional robustness. The user is presented with veering-correction speech cues and feedback about performance. While this system seemed promising, it did not undergo a thorough evaluation.

Panëels et al. [13] built on the AVTD with their *Walking Straight* application, which also uses the gyroscope to measure body sway and orientation. The experiment consisted of walking 15 m in a straight line towards a target after initially being positioned in the correct orientation. Their testing was conducted in a controlled outdoor environment, rather than at an actual intersection. They found that the system reduced veering to half that observed during the control condition. The authors also focused on the feedback modality, recognizing that mobility training teaches blind individuals to move away from a sound. As such, they concluded that the most effective method for providing veering feedback was a continuous beep rendered in the ear on the side to which the user was veering.

While the systems described above may be effective in an ideal setting, sensor stability can prove problematic when it is used continuously. For example, Blum et al. [37, 35] demonstrated the instability of orientation sensors when using the magnetometer (smartphones in 2012), finding that the mean heading error can reach 30^o in areas with high-rise buildings, making them entirely unreliable for the street-crossing task. More recently, Mohssen et al. [54] evaluated a system that integrates the output of the gyroscope with the magnetometer to obtain improved orientation of an Android smartphone, and still found errors of 15°, which would still be beyond the safety margins for the street-crossing task. These errors force smartphone applications to operate under the assumption that the user will be properly oriented at the onset of crossing. As noted in previous work [34], this is not always the case.

2.1.2 Vision-Based Systems

Relying instead on visual information provided by a smartphone camera presents an attractive alternative to non-vision based sensing. This is especially the case considering that non-visual understanding of the environment is not only less effective and less efficient, but also potentially dangerous, compared to scanning the surrounding using vision [55]. However, processing the wide variety of street scenes to extract the appropriate features, if present, for guidance has long been a daunting challenge.

Shen et al. [31] developed a prototype smartphone application, utilizing the camera for detection of zebra crossings using segmentation of the edges of the stripes in the scene. Taking advantage of improved camera systems in more recent mobile phones, Ahmetovic et al. [56] developed a system that uses a five-step process, computing the position of the zebra-crossing by using a combination of the camera and the device's accelerometer as inputs. The user detects and crosses an intersection by holding and panning the mobile phone around, with the camera "looking" for the zebra-crossing. The output of the accelerometer informs the user of the position of the phone relative to the ground. The results were promising, with all the subjects successfully capable of crossing a 6 m road in an average 3-5 s. One of the major limitations of such systems is their reliance on the zebra pattern, precluding use for intersections that employ different markings. To overcome this limitation in part, Ahmetovic et al. [57] mined existing image databases (e.g., Google Street View images) to plan a route that only traverses zebra crossings, a solution that is obviously inapplicable for many cities. Another limitation of such a system is its inability to deal with partial occlusions due to erased markings or snow-covered crossings, which can direct users to move in the wrong direction, leading to potentially dangerous situations.

Ivanchenko et al. [58] developed *Crosswatch*, which detects the more common two-stripe crosswalk. They utilize computer vision techniques to model the stripes, providing audio feedback to inform the user when the system detects at least one stripe of the crosswalk and to inform the user when their feet are inside the two-lane corridor. In addition, they use accelerometer readings to estimate the direction of gravity, making it easier for the user to position the camera in the correct orientation. The preliminary experiments required that the blind user correctly identify the location of a crosswalk. However, the reported experiments did not include the actual task of crossing the intersection and, therefore, do not allow for conclusions as to the effectiveness of their solution. Additionally, it is not clear how such a system would handle partially or fully occluded stripes. Finally, from the user's perspective, after multiple discussions and observational sessions with visually impaired individuals, we found that they prefer to keep their hands free, which is not possible with the Crosswatch system.

Poggi et al. [59, 60] designed a pocket-sized device with an embedded CPU, coupled with a custom depth-sensing camera attached to wearable glasses, to assist with initial orientation of the user prior to crossing at an intersection. This device used the dense disparity map from the depth-sensing camera to determine the ground plane, which served as a reliable way to discriminate between the ground and the rest of the view in the image. They also trained a CNN model, similar to the LeNet architecture [1], which takes as input a warped image of the ground and, if a crosswalk is present, determines its orientation. The authors obtained near-perfect accuracy on their test set, a testimonial to the power of these models, although they did not report on any on-line testing to validate the system in the real world. It is unclear whether the system can easily be adapted to provide users with real-time veering feedback, beyond the initial alignment at the start of the intersection crossing.

Instead of relying on the availability of certain features at crosswalks, this thesis evaluates the use of convolutional neural networks to develop an application that assists people with visual impairments in crossing intersections with minimal veering. The challenge with CNN models is their requirement of large amounts of data in order for them to converge to good results. However, previous work [61] has shown that the features CNN models learn can be transferred to tackle problems from different domains. This work allows the adaptation of models that achieved state-of-the-art performance on the Imagenet dataset [62] to smaller datasets while preserving the performance they achieve.

2.2 Assessment Strategies of Captioning Models

There are two main approaches used in the literature to evaluate the quality of image captions. The first and most prominent method is the use of automatic evaluation metrics, which generally use a measure of similarity between the human-generated captions and the ones obtained by the model to calculate the accuracy of the model. The second approach is to employ human evaluators, whether in the lab or crowdsourcing, who are given an image with the corresponding caption and are asked to score the accuracy of the model. In this section, we present the various evaluation methods used in the literature for both approaches.

2.2.1 Automatic Evaluation Metrics

With the surge of models generating captions for images, researchers have looked for ways to automatically evaluate the performance of their models. Previous works [2, 22, 23] in image captioning primarily use automatic assessment metrics as a way of evaluating their models. These include the BLEU score [47], ROUGE [63], METEOR [64], CIDEr [65] and SPICE [6]. We provide a brief description of these metrics in Table 2.1.

Metric	Description
BLEU [47]	One of the first metrics proposed for measuring similarity between two sentences, it is defined as the geometric mean of n-gram precision scores, i.e., a measure of how many words in the machine generated caption appear in the reference ground truth captions. It also has a penalty for brevity of the caption.
ROUGE [63]	As opposed to the BLEU score which measures precision, ROUGE measures the n-gram recall score, i.e., how many words in the ground truth caption appear in the caption provided by the model.
METEOR [64]	Originally proposed to address several weaknesses observed in the BLEU score, this metric computes the harmonic mean of precision and recall using the n-gram representation of the descriptions, automatically placing a penalty on brevity as explained by Banerjee et al. [64]. It also uses WordNet-based synonym matching to allow for some drift from the ground truth.
CIDEr [65]	A recently proposed metric specifically designed for evaluating the quality of image descriptions. It does so by measuring the consensus between a candidate description with the set of ground truth sentences for a given image, e.g., in Microsoft COCO, each image has five ground truth captions. The measure is calculated using n-gram representations of the captions.
SPICE [6]	The most recently proposed metric for image caption evaluation, this metric uses scene graphs to determine the accuracy of a caption. This is accomplished by first computing the scene graph of the ground truth and the generated captions, and calculates the F-score to evaluate the similarity between the graphs.

Table 2.1: Brief description of the most popular automatic assessment metrics used for the image captioning task. We invite the reader to read the papers for more details.

The prominence of these metrics can be attributed to the fact that, excluding SPICE, they correspond to the ones used on the official Microsoft COCO evaluation server [44]. However, as noted by Anderson et al. [6], there are a few problems with using these evaluation metrics to evaluate image captioning models. Firstly, BLEU, METEOR and ROUGE were primarily designed for the machine translation community, and therefore are not good indicators of correctness on a vision-to-language task. In addition, even though it was designed for the image captioning task, CIDEr [65] is based on n-gram similarity metrics, meaning the output score can be tricked by having similar combination of words. For example, taken from Anderson et al. [6], "A young girl standing on top of a tennis court." and "A giraffe standing on top of a green field." would produce a high similarity score due to the 5-gram "…standing on top of a…"

More problematic is the fact that none of the four metrics have shown good correlation with human judgment. As detailed in the human evaluation study performed by Anderson et al. [6], SPICE significantly outperforms the other metrics, achieving the highest correlation with human judgments on the Microsoft COCO dataset. In addition, as part of their study, they found that only SPICE rewards detailed captions, with BLEU and ROUGE penalizing the score for too much details. Important to note that SPICE is the only one to rank human generated captions first, with CIDEr and METEOR ranking them 7^{th} and 4^{th}, respectively. Their findings are aligned with Elliot and Keller's work [66] who found that METEOR, BLEU and ROUGE were all weakly to moderately correlated with human judgment.

At the time of writing, we are not aware of an automatic evaluation metric that provides measures of relevancy geared towards the visually impaired community. As we explore in Chapter 5, the needs of visually impaired individuals differ greatly from those of sighted individuals as spatial awareness becomes an important aspect of the description. We can see that from the analysis presented above by Anderson et al. [6], SPICE seems to be the best automatic assessment metric for the image captioning task, having good agreement with human judgment of captions. It would be interesting to evaluate a model that generates descriptions for visually impaired people using this metric, providing researchers with insight on how well the captions align with a blind user's detail requirement. A possible outcome of our work is the creation of an application that results in a scene description dataset for the blind community. If such a result is achieved, the use of SPICE to evaluate models trained on this dataset would be an approach to assess the amount of detail and relevancy of captions.

2.2.2 Human Evaluation of Image Captions

In this section, we review previous work that employed sighted individuals to evaluate the correctness of various captioning models. We also present their captioning method at a high level, giving the reader an overview of the design of such models.

Vinyals et al. [2] proposed one of the first approaches that leverages the power of deep learning producing an end-to-end system tackling the image captioning problem. Their model consisted of a CNN that encodes the image into a vector that is then read by an LSTM, a type of recurrent neural network, to produce a caption word by word. Since the authors have made their code available, we use this model in our framework to evaluate its effectiveness at providing captions for the blind community. The authors evaluate their approach using the BLEU, METEOR and CIDEr metrics, achieving state of the art performance compared to previous methods. In addition, they perform a human evaluation step where raters are asked to rank image descriptions on a scale of $1 - 4$ based on how well they match the image. This was accomplished using Amazon Mechanical Turk where two workers were employed to rate the captions, having an average agreement level of 65%. In this evaluation, 1 corresponded to “Unrelated to the image”, 2 to “Somewhat related to the image”, 3 to “Describes with minor errors” and 4 to “Describes without errors”. On a 1000-image subset from Microsoft COCO’s test set, they produced an average score of 2.72. Moreover, the authors also evaluated on the Flickr8k test set obtaining an average score of 2.37 compared to the ground truth average score of 3.89. The authors note that their results show that BLEU and other metrics are not perfect metrics for this task. Note that this work was published before the introduction of the SPICE metric.

Fang et al. [67] proposed an approach that first uses weakly supervised learning to create detectors for different sub-regions of an image, then produced sentences that are *likely* given the detections from the first stage, and finally ranks the sentences using a Deep Multimodal Similarity Model, resulting in the most likely description for a given image. Using Mechanical Turk workers from CrowdFlower, which avoids spammers, they perform a human evaluation of their model where each worker is presented with an image and two captions. One of the captions is automatically generated by the model while the other is chosen from the dataset. The worker is asked to select the caption that better describes the image or to choose a “same” option when they are equal in quality. Their experiment consisted of having 250 humans compare 20 caption pairs each, with five humans judging each caption pair. They found that 34% of the descriptions produced by their model were judged to have the same or better quality than the human-written descriptions.

Tran et al. [68] hypothesize that while previous methods (such as that of Vinyals et al. [2]) perform well on automatic assessment metrics in the ideal experimental setting with test images collected in controlled environments, it is unclear how they would perform on open-domain images. They also note that most of these image captioning systems describe generic visual content without identifying key features such as landmarks or celebrities. Their model builds on the model from Fang et al. [67] by enriching the initial phase of detection to include broader visual contents. In addition, they build a model that estimates the confidence score for a caption output based on the vision and text features, providing a “back-off” caption for difficult cases. Using CrowdFlower, they perform a series of human evaluation experiments where evaluators are asked to rate images as *Excellent, Good, Bad* or *Embarrassing.* The authors compare their model to that of Fang el al. [67] on images from Microsoft COCO’s test set as well as images collected from 100 popular Instagram accounts, providing the images from the wild. On the COCO test set, they obtained 51.8% excellent rating compared to 40.6% from [67], and obtained 25.4% versus 12.0% of [67] on the Instagram test set. Their proposed model resulted in the release of CaptionBot [5].

Dai et al. [42] proposed another image captioning model based on generative adversarial networks (GANs) [69] with the goal of avoiding the use of the maximum likelihood objective that typically results in generic and repetitive captions. This is accomplished by employing the discriminator (a second neural network) portion of GANs as an evaluator of the caption, with its goals being to obtain descriptions that are natural and semantically relevant to the image. To assess their model, they use the metrics we describe in Section 2.2.1 as well as a user study with qualitative comparison. Their user study consisted of 30 human evaluators who are each presented with an image and two sentences, and are asked to choose the better description for the image. This method of choosing the better of two captions was also previously explored by Devlin et al. [70] where they analyzed different language models for captions given the same image feature extracting CNN. In [42], the origin of the descriptions varies as they not only compare their model to ground truth captions, but also to the model presented by Vinyals et al. [2]. They found that their model was better than or equivalent to the one proposed by Vinyals et al. 61% of the time, and that their model was better than or equivalent to the ground truth 24% of the time while [2] was better 9% of the time. Interestingly, when using the automatic assessment metrics, the model proposed by Vinyals et al. [2] consistently performed better than both the proposed model and the human ground truth (except on SPICE, where it came in 2^{nd} after humans). This further shows the deficiencies of the automatic assessment metric we presented in Section 2.2.1.

Aditya et al. [71] present a model that integrates deep learning based vision, which detects objects, scenes and constituents, with concept modelling of commonsense knowledge constructed from text. The authors use commonsense knowledge acquired from image annotations combined with a Bayesian network to capture how objects interact in the scene. This output of their model is a Scene Description Graph (SDG) that depicts how different entities relate to and interact with each other. The SDG can then be used for further reasoning about the scene and to generate captions. After turning this model to a caption generator, the authors perform a human evaluation qualitative study on their captions using Amazon Mechanical Turk. Evaluators were asked to rate the image captions based on relevance, i.e., how much the description conveys the image content, as well as thoroughness, i.e., how much of the image content is conveyed by the description. This was done using a discrete scale ranging from $1-5$, where 1 indicated low relevance or low thoroughness, and 5 indicated high relevance or high thoroughness.

In their assessment of their proposed SPICE metric, Anderson et al [6] performed a human evaluation study to decide the correlation between different metrics (including SPICE) and human evaluation. As we already discussed various parts of the assessment of their metric, here we would like to focus on the human evaluation portion. In rating the captions, evaluators answered five questions:

1. Choose the better caption between two captions.
2. Choose which caption was machine generated, i.e., passing the Turing Test.
3. Rate the correctness of the caption on a scale from $1-5$ (incorrect - correct).
4. Rate the amount of detail contained in the caption on a scale from $1-5$ (lacking details - very detailed).
5. Decide whether the machine-generated caption captures the same idea as the human one.

Here, 1 and 2 provided a measure of the caption's overall quality, 3 measured its correctness, 4 measured the amount of detail and 5 measured its saliency. In our opinion, their method is the most complete evaluation guidelines one should follow for the assessment of image captions.

Chapter 3

Technical Preliminaries

In this Chapter, we describe the technical preliminaries that enabled the design of our intersection crossing assistant presented in Chapter 4. Next, we present the approach of Vinyal et al. [2] who integrate deep learning models to generate natural language descriptions of images. We review the basic theory of imitation learning, a learning technique we use to formulate our street crossing model. Finally, we present the Kalman filtering algorithm employed in our final street crossing mobile application.

3.1 Deep Learning Architectures

Most of the content presented in this section is summarized from Chapter 6 of the Deep Learning textbook by Bengio et al. [72]. We refer the reader to this chapter for more information about feed-forward neural networks.

Deep learning architectures have produced state-of-the-art results in many application areas such as image classification [16], object detection[20] and machine translation[73]. The basic building block of these models are multi-layered perceptrons (MLP), which have been widely used in supervised and unsupervised learning problems.

In supervised learning, given a dataset consisting of an input X and a target output Y, the goal is to obtain $f(x)$ that matches $f^*(x)$, the true data generating function. The training data provides noisy approximates of $f^*(x)$, i.e., given an input x, the model should output a value close to y. In an MLP, the non-linear function f is defined by its learnable parameters θ, and the goal is to learn θ such that $f^(x;\theta) = f^*(x)$. We show an example of an MLP with a single hidden layer in Figure 3.1. The trainable network parameters θ are comprised of all layer parameters, e.g., $W^{(1)}$, $w^{(2)}$, $b^{(1)}$ and $b^{(2)}$ from the example MLP of Figure 3.1.

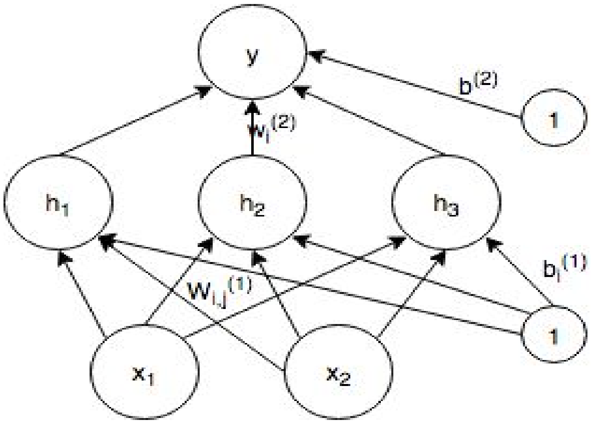


Fig. 3.1: Example of a multi-layered perceptron with a single hidden layer.

An MLP is formulated by the composition of multiple non-linear functions. For example, a network with n layers can be formulated as $f(x) = f^{(n)}(f^{(n-1)}(\ldots f^{(1)}(x)\ldots))$ where each layer $f^{(j)} = g(\mathbf{w^{(j)}h^{(j-1)}} + \mathbf{b^{(j)}})$. It is also known as a feed-forward neural network because information flows from x, through the intermediate computations defined by f, and to the output y without any feedback connections. Networks with feedback connections are known as recurrent neural networks (RNN), and are explained in Section 3.1.2.

Hidden layers in an MLP allow learning of non-linear transformations of the input data. As the hidden units do not have a label, the learning algorithm must decide the network parameters θ that produce the desired output y. The transformed view of the data, $\phi(x)$, can be viewed as a set of features describing the input x, providing a new representation. Theoretically, a feed-forward neural network with a single hidden layer can represent any function [72]. However, that layer may be too large as well as fail to learn and generalize to the data distribution. Instead, increasing the number of layers provides more layers of abstraction of the input data. From a computational perspective, it is more efficient to include depth in a network than width as the number of hidden units required by a "shallow" model is exponential in the number of datapoints.

A deep network's ability to learn depends on the type of non-linear activation function (the function $g(.)$ above) used. Without a non-linear activation function, an MLP would have the capacity to only learn functions for linearly separable data [72]. Many activation functions have been proposed in the past and it is still an active area of research, according to Bengio et al. [72]. The most popular one previously used is the Sigmoid function, shown in Equation 3.1, which produces an output bounded between 0 and 1.

$$g(z) = \frac{1}{1+e^{-z}} \tag{3.1}$$

The hyperbolic tangent function, shown in Equation 3.2, produces an output bounded by -1 and 1.

$$g(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \tag{3.2}$$

Their use in feed-forward neural networks has become less prevalent, and even discouraged, as they saturate to their high value when z is very high or to their low value when z is very low, making learning very difficult when using a gradient-based approach. The Rectified Linear Units (also known as $ReLU$) activation function, shown in Equation 3.3, was proposed to alleviate these problems.

$$g(z) = max\{0, z\} \tag{3.3}$$

Unlike the other two methods, the ReLU function is linear in one portion of its input with a slope of 1, meaning that the derivatives through it remain large whenever the unit is active, i.e., $z > 0$. The introduction of ReLU, along with better Graphical Processing Units, has allowed efficient learning in models with many hidden layers, hence the word "deep" in deep learning.

The output layer of an MLP also has an activation function, the form of which depends on the task at hand. For a classification task, the number of neurons in the output layer is the number of classes of the dataset labels y, and the goal is to predict the correct class using a Softmax distribution layer, which follows the form in Equation 3.4. This distribution provides a real value between 0 and 1 representing the probability of each class. The predicted class is chosen as the class with the highest probability, i.e., an *argmax* over the output vector of the softmax layer.

$$g(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{3.4}$$

For a regression task, the goal is to approximate a real-valued quantity, and is typically obtained by having a linear output of the final layer composed of a single neuron. The form of the final layer is shown in Equation 3.5

$$g(z) = \mathbf{w}^T\mathbf{z} + \mathbf{b} \tag{3.5}$$

To train an MLP, we need to consider a cost function that minimizes the discrepancy between the output and the target. In modern neural networks, the maximum likelihood objective is used to train neural networks. The cost function, shown in Equation 3.6, is therefore the negative log-likelihood of the function, which can equivalently be expressed as a measure of the cross-entropy between the training data and the model distribution.

$$J(\theta) = -\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{data}} \log[p_{model}(\mathbf{y}|\mathbf{x})] \tag{3.6}$$

For a classification task, it is sufficient to use the output Softmax distribution directly in Equation 3.6 to evaluate the current loss of the model. For a regression task, expanding Equation 3.6 will result in minimizing the mean squared error, shown in Equation 3.7.

$$J(\theta) = \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{data}} \parallel \mathbf{y} - f(\mathbf{x};\theta) \parallel^2 \tag{3.7}$$

Finally, training of a feed-forward neural network is accomplished using the *backpropagation* algorithm. This algorithm describes how one should update the parameters θ to minimize the cost function $J(\theta)$ using gradient descent. For each layer $f^{(j)} = g(\mathbf{w^{(j)}h^{(j-1)}}+\mathbf{b^{(j)}})$ in the network, the gradients of the cost function are first computed with respect to the layer's output $f^{(j)}$, and the gradient of the layer output $f^{(j)}$ is computed with respect to the parameters θ_j, where $\theta_j = \{\mathbf{w^{(j)}}, \mathbf{b^{(j)}}\},$. Then, using the chain rule, the gradient of the cost with respect to the parameters of that layer is obtained, shown in Equation 3.8.

$$\nabla_{\theta_j} J = \frac{\delta J}{\delta f^{(j)}} \nabla_{\theta_j} f^{(j)} \tag{3.8}$$

The parameter update rule is shown in equation 3.9, with α being the learning rate.

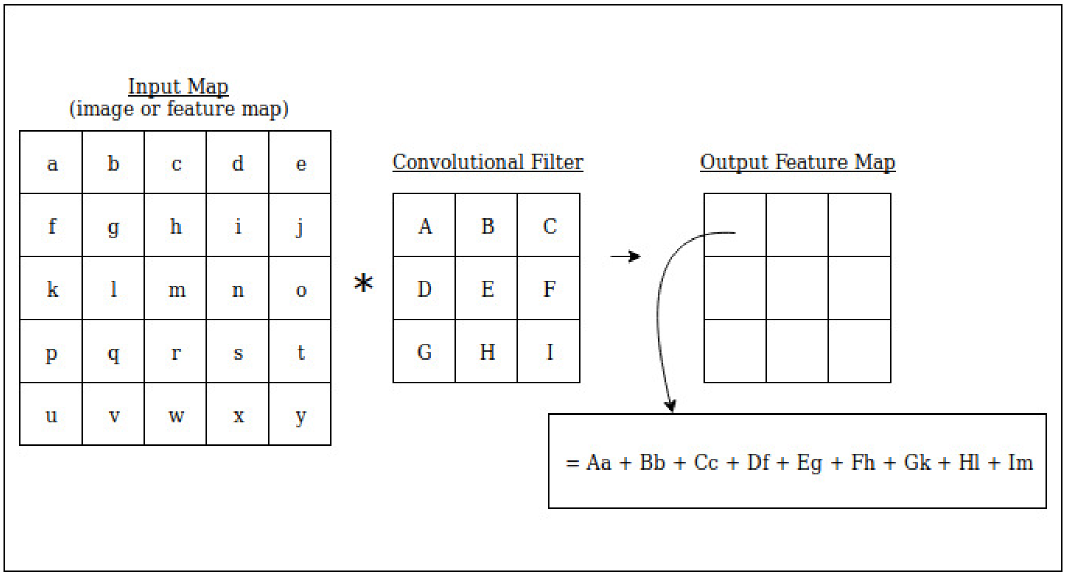
$$\theta_j \leftarrow \theta_j - \alpha \nabla_{\theta_i} J \tag{3.9}$$

Both convolutional neural networks (CNNs) and recurrent neural networks (RNNs) build on feed-forward neural networks, and are geared towards specific types of data. In the following sections, we delve into the technical details of these models. In addition, we present an integration of these models that results in an image captioning model trainable in an end-to-end fashion.

3.1.1 Convolutional Neural Networks

An image is a high-dimensional form of data, and can be thought of as a 2-D grid of pixels with a certain number of color channels (1 channel for grayscale or 3 channels for RGB colors). It is possible to reshape this data into a column vector and design an MLP to process this form of data. However, as the dimensions of the image increase, the number of parameters in the MLP would become computationally infeasible. Convolutional neural networks (CNNs) are a specialized type of neural network designed to efficiently and effectively handle grid-like data, such as images. The name "convolutional neural network" is given to any feed-forward neural network that has a layer employing the *convolution* operation instead of the general matrix multiplication presented above.

The primary objective of the convolution operation is to extract features from an input image. The operator preserves the spatial relationship between pixels by learning features over square portions of the input. A kernel, known as the convolutional filter, is slid across the input, and an element-wise multiplication with the filter weight matrix followed by a sum produces the output feature map. This operation is shown graphically in Figure 3.2, where the convolutional filter values are the learned parameters of the neural network.



a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y

A	B	C
D	E	F
G	H	I

= Aa + Bb + Cc + Df + Eg + Fh + Gk + Hl + Im

Fig. 3.2: Example of the application of a 3×3 convolutional filter over a 5×5 input image. The output, named a feature map, is 3×3 since the stride is 1 and no zero padding is used.

The convolution operation has several hyperparameters associated with it. For instance, one can control the **stride** with which the convolution is applied, i.e., the number of pixels by which the filter is slid over the input matrix. Another hyperparameter is the amount of **zero-padding** applied to the input Map, which allows the filter to be applied to the bordering elements of the input map. In the example shown in Figure 3.2, a stride of 1 is used and no zero padding is applied. The final, and perhaps most important, hyperparameter is the **depth**, or number of filters, applied per layer. Typically, the number of filters applied to an input is on the order of 64. The output feature maps are stacked along the third dimension along the channel axis. In the example of Figure 3.2, if we used 64 filters each being 3×3 in size, the output feature maps would have a dimension of $3 \times 3 \times 64$.

There are interesting properties associated with the convolution operator. In the MLP model, every output unit in a layer interacts with every input unit since the weight matrix contains a separate parameter describing every interaction. In contrast, the convolution operator has **sparse interactions** accomplished by having the filter smaller than the input map, i.e., each output node in a feature map is connected to a specific group of input nodes. The advantages associated with sparse interactions are a reduction in the memory requirements of the model, improved statistical efficiency, and a reduction in the number of operations in computing the output. In addition, each element of the weight matrix in an MLP model is used exactly once during the computation of the output layer, making it expensive in storage requirements. The convolution operator overcomes this via **parameter sharing**, achieved by the sliding of the convolutional filter over the input map. A consequence of sharing the parameters across different areas is that the value of the weight applied to one area is *tied* to the value of the weight applied in another area. Therefore, we learn a single set of parameters for the whole input rather than learn a separate set of parameters for every location. This makes sense in the image domain since we expect the occurrence of specific features in one area to be equally likely to appear in another area. This is also known as **equivariance to translation**, i.e., if the input map changes, we expect the output map to change in the same way.

Typically after a convolutional operation, the output feature maps are put through a non-linearity function such as the ReLU layer (see Equation 3.3). Many models then put the output of the non-linearity through a *pooling* layer. Pooling layers operate over a certain location and replace the values with summary statistics of the nearby outputs. For example, the **average pooling** operation computes the average of a rectangular area in the non-linearly transformed feature maps and provides a new feature with values of the average. In general, pooling operations help make representations approximately invariant to small translations of the input. Some literature groups the convolution operator with the non-linearity layer and the pooling layer into one, which we refer to as *CONV-NLIN-POOL*.

In a typical CNN, there are a few rounds of *CONV-NLIN-POOL* layers, followed by fully-connected layers, and a final output layer which depends on the task at hand, whether classification or regression. We present one of the first convolutional neural networks ever proposed in Figure 3.3.

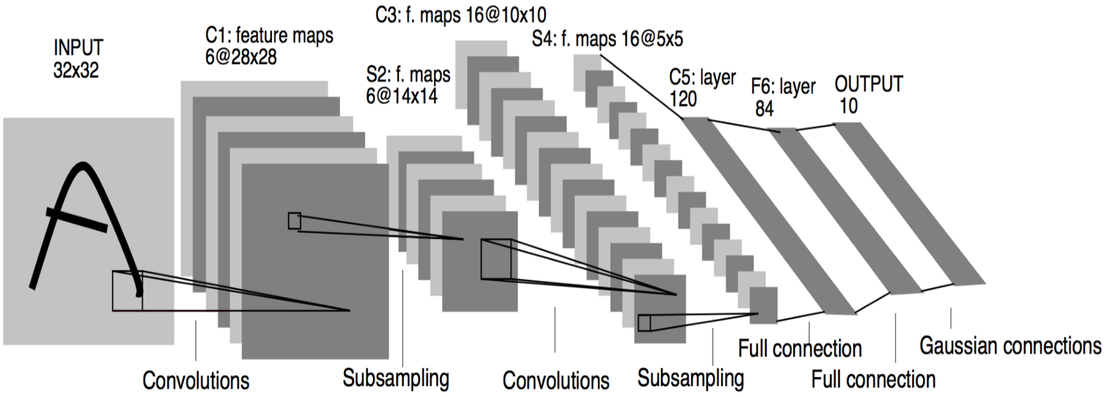


Fig. 3.3: Example of the LeNet CNN architecture proposed by Lecun et al. [1]. Note that "subsampling" is another way of saying pooling.

3.1.2 Recurrent Neural Networks

Recurrent neural networks (RNNs) are specialized models designed for sequential data $x_1, \ldots, x_\tau$, such as data from the natural language domain. This is accomplished through feedback loops in the network, allowing it to hold information about all previous inputs from x_1 to x_{t-1}. Just as CNNs are efficient for processing images of increasing size, RNNs are well-equiped to handle variable-length sequences of data. Figure 3.4 shows the directed cyclic graph representing an RNN.

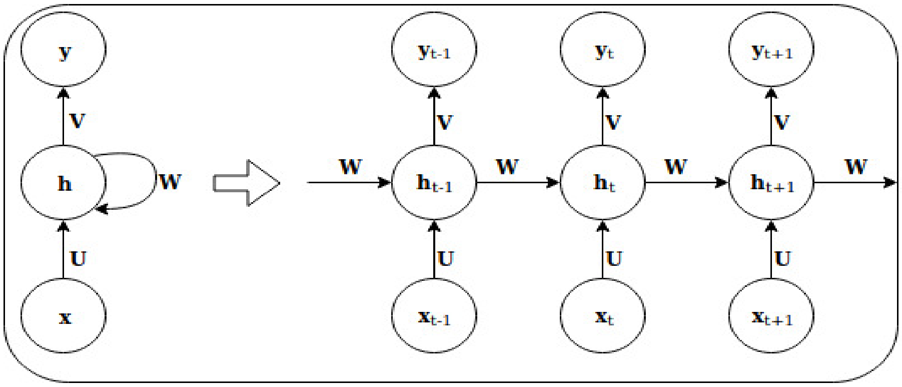


Fig. 3.4: A directed cyclic graph representing a recurrent neural network.

The dot product $\mathbf{U^T x_t}$ (usually followed by a non-linear activation function) is a way for the hidden layer to extract important information from the input. $\mathbf{W^T h_{t-1}}$ provides the network a way of extracting information from the entire history of past inputs $\mathbf{x_1}$ to $\mathbf{x_{t-1}}$. For this reason, the hidden state can also be viewed as the memory of the network as it holds information about previous inputs. This information, along with the information obtained from the current input $\mathbf{x_t}$, is used to calculate the new hidden state $\mathbf{h_t}$. Finally, the dot product $\mathbf{V^T h}$, followed by an activation layer, is trained to produce the desired output $\mathbf{y_t}$ using maximum likelihood.

As we can see from the diagram of Figure 3.4, the weights are shared across timesteps. Th use of parameter sharing allows RNNs to extend and generalize to examples of different lengths. This works well for sequential data since we are typically performing the same task at each timestep, only with different inputs. Similar to CNNs, the sharing of parameters results in a more time and storage efficient graph as less parameters need to be learned compared to standard feed-forward neural networks.

Recurrent neural networks and their variants have been widely used in sequential data domains, e.g., machine translation, text generation, language modelling, speech recognition. In the next section, we present an image captioning application that employs CNNs combined with RNNs. As we do not directly employ RNNs in this thesis, we refer the reader to Chapter 10 of the deep learning textbook by Bengio et al. [72] for more details regarding training and extensions of the basic architecture shown in Figure 3.4.

3.1.3 Combination of Recurrent Models with CNNs

Since we discuss image captioning models in this thesis, it is important that the reader knows one previously proposed deep learning model to accomplish this task. Motivated by work in machine translation, Vinyals et al. [2] proposed an **encoder-decoder** architecture, shown in Figure 3.5. In this model, a CNN (encoder) is used to compute features that are used as the initial input to an RNN (decoder). The RNN is then tasked with generating each word of the caption.

The CNN they use is the *Inception-V3* model proposed by Szegedy et al. [74], which was pre-trained on the Imagenet classification dataset[62]. As we explore in Section 4.1.4.1, using a pre-trained model for image classification provides an effective image feature extractor. In CNNs, this is accomplished by removing the final classification layer (and perhaps more layers from the end), and using the output vector as the new representation of the image.

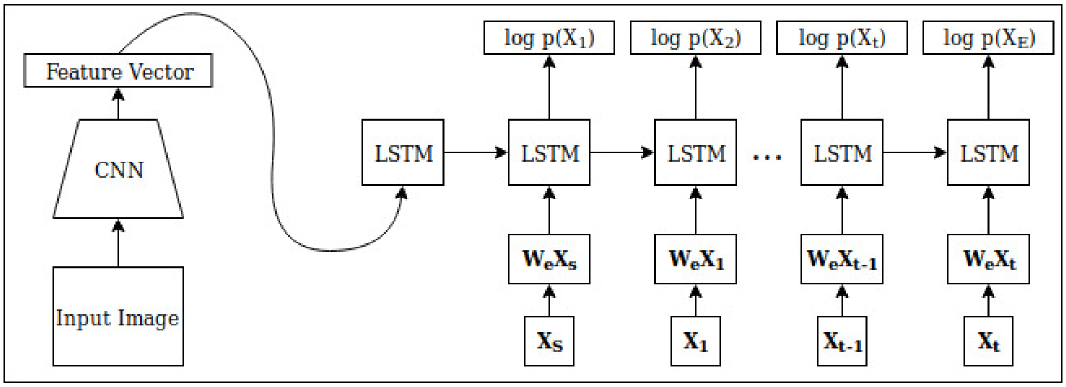


Fig. 3.5: Image captioning model proposed by Vinyals et al. [2].

In their work, the authors use a more efficient RNN named the Long-Short Term Memory (LSTM) proposed by Hochreiter et al. [75] to overcome many of the challenges faced when training a standard RNN. The decoder of their architecture, i.e., the LSTM, obtains the encoded *feature vector* of the image, and produces the initial hidden state h_0. At the next timestep, the LSTM obtains the *starter-token* word (shown by X_S in Figure 3.5), which informs the model to start producing an output. Every word, including the starter-token, is encoded through a pre-trained word embedding matrix (W_e) producing a vector representation of the word. After passing through the LSTM layer, the decoder outputs a probability distribution over the vocabulary of words in the dataset. The word with the highest probability is the predicted word of the model. In the caption generation stage of the model, i.e., after it has been trained, the predicted word is used as the input in the next timestep. During training, the input word at the next timestep is the target caption's next word obtained from the dataset. Once the end-of-sentence token (shown by X_E in Figure 3.5) obtains the highest probability in the output, the generation process is complete. This model is trained using the maximum likelihood objective, minimizing the classification loss between the output probability distribution and the target correct word in the caption viewed as a one-hot vector. Here, the one-hot vector is a sparse vector with a dimension equal to the number of words in the vocabulary of the dataset, and a value of 1 at the index of the target correct word, 0 otherwise.

3.2 Imitation Learning

Imitation learning, also known as Learning from Demonstration (LfD), is based on the idea of transferring human behavior to intelligent agents [76] [77]. Employing this learning approach provides practitioners with an intuitive way of teaching agents a certain task as humans communicate knowledge in this way. The main goal of imitation learning is to implicitly give the agent prior information about the task at hand. This is accomplished by showing it examples of human behaviour while performing the task, and having the agent analyze the data to extract the *policy* humans follow.

An agent's policy is a function that maps every state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$, i.e., $\pi : \mathcal{S} \rightarrow \mathcal{A}$. Conceptually, algorithms in the LfD domain aim to acquire the optimal policy π^* for a task from a series of demonstrations $\mathcal{D} = \{d_1, d_2, ..., d_m\}$ that can guide an agent while autonomously performing the task. To completely formulate an LfD solution, one must establish the structure of the world states $s \in \mathcal{S}$ that the agent may reach, the actions $a \in \mathcal{A}$ that the agent is capable of performing, and a transition function $\mathcal{T}(s'|s, a)$ that expresses the probability of landing in $s' \in \mathcal{S}$ given that the agent executes action a from state s. In most real-life scenarios, the state is not fully observable. Most LfD models handle this uncertainty by relying on the agent's observations of the world, $z \in \mathcal{Z}$, instead of the complete representation of its internal structure [77]. Therefore, our LfD method must determine the optimal policy, $\pi^* : \mathcal{Z} \rightarrow \mathcal{A}$ using demonstrations $d_i = (z_i, a_i) \in \mathcal{D} : \mathcal{Z} \times \mathcal{A}$.

There are many application areas that have employed imitation learning. These include *autonomous vehicles*, such as learning to fly an aircraft from demonstrations provided via remote control, or self-driving road vehicles. Other areas include *Humanoid robots*, which allows robots to replace some of the workload that humans do, and *electronic games*, which can allow an enhanced game experience and immersion in games. Relevant to our work is the area of *assistive robotics*, such as intelligent robots learning to help elderly people with daily activities or help recovering individuals with daily activities. Assistive agents can also help with sociological and mental problems. For such agents to be useful in social contexts, their behaviour should be human-like allowing the person being assisted to recognize the robots behaviour, providing further motivation for the use of imitation learning. In Chapter 4 of this thesis, we gather the knowledge of sighted "experts" on the intersection-crossing task, and transfer it to an intelligent assistive agent for the visually impaired using convolutional neural networks.

3.3 Kalman Filter Algorithm

Kalman Filtering [78] is an algorithm that produces estimates of unknown variables using a series of noisy measurements observed over time. These filters have been applied in numerous application domains such as in navigating and controlling autonomous vehicles.

There are two steps in Kalman filtering: the prediction step and the update step. The first part of the prediction phase, shown in Equation 3.10, decides what the current state should be based on the previous state, the transition function of the system, and the control action taken.

$$x' = Fx + u \tag{3.10}$$

where x is the object state. F is the state transition matrix, i.e., a matrix that transforms the previous state x. u incorporates information about external actions that may affect the system. x' is the predicted object state according to the state transition matrix F and the action u taken.

The second part of the prediction phase, shown in Equation 3.11, provides an estimate of the uncertainty in the predicted state x'.

$$P' = FPF^T + Q \tag{3.11}$$

where P' is the predicted object uncertainty. P is the uncertainty of the object's state, and is related to error in the measurements. Q is the uncertainty of the process.

The update phase takes measurements from the input sensors, including the noise of the sensor, and provides a new estimate of the state. The first step is to calculate the Kalman gain using Equation 3.12. This gain informs the algorithm of how much we should let the new measurements effect our prior belief in the state, as calculated in Equation 3.10.

$$K = P'H^T(HP'H^T + R)^{-1} \tag{3.12}$$

where H is a model of the sensor used to provide the measurement, and is typically difficult to determine its value. H can also be viewed as transforming our prediction x into the space of our measurements. R represents the uncertainty of the sensor measurements. K is the Kalman gain value, with smaller values signifying the prior beliefs are less affected by new measurements.

Next, we calculate the new object state based on the new measurement from our sensor, the Kalman gain, and the predicted state from Equation 3.10. This calculation is performed in Equation 3.13.

$$x = x' + K(z - Hx') \tag{3.13}$$

where z is the input measurement obtained from the sensor.

The last step is to update the process uncertainty P using the Kalman gain. This update is calculated using Equation 3.14. Intuitively, P decreases with every update by an amount that depends on our certainty of the measurement.

$$P = (I - KH)P' \tag{3.14}$$

As an example, we explain at a high-level how a Kalman filter can be applied to determine the location and velocity of a vehicle we are tracking. The vehicle's state x is a vector holding its location and velocity, both in two dimensions. The state transition matrix F represents the model of the vehicle in the world, and is used to predict the next location and velocity. The action u is an action that a controller (such as a driver) has taken, e.g., braking. The uncertainty in the state P represents how sure we are of the measurements we obtained from the sensors. The process uncertainty Q captures the mistakes in our assumptions of F, such as the car never accelerating when in reality it might. If our location measurements are in centimeters while our state units are in meters, H converts the state values into the measurement units. R is the error in the obtained GPS coordinates.

Chapter 4

Assisting Blind People with Intersection Crossing

Despite the recent surge of work in intelligent robotics, to our knowledge, the results from this research field have scarcely been applied to alleviate sensorial, motor and cognitive impairments in humans [79]. We believe that such research, in particular the technique of learning from demonstration in imitation learning, is well suited to addressing the problem of veering during street crossing. In this chapter, we present our solution to the street crossing challenge for blind individuals. We present the methodology followed in preparing the model and the quantitative assessment of the model in the lab. Next, we present the motivation behind many design choices of our mobile application. Following the completion of the mobile application, we conduct an experiment with visually impaired participants to assess the effectiveness and feasibility of the proposed solution. Finally, we conduct a series of iterative experiments with a visually impaired volunteer to improve our mobile application incrementally, preparing for future work that will perform a larger scale experiment of the application.

4.1 Methodology

In this section, we describe our formulation of the problem as an imitation learning problem. Furthermore, we describe the data acquisition strategy as well as the policy derivation method.

4.1.1 Task Demonstrations

To reproduce the veering behavior that may be exhibited when crossing, we first familiarized ourselves with the crossing behaviors of visually impaired individuals through two observation sessions, which found the following:

- Visually impaired individuals rely on a constant flow of traffic to orient themselves accurately.
- Even with the help of their mobility training, guide-dog, or cane, orientation can be problematic when traveling through environments with low traffic flow, such as suburban areas. This can result in long wait times at intersections.
- As deviations into the traffic flow are a worst-case scenario, individuals with vision impairments tend to veer to the interior side of the crossing.

We divided our collection of demonstrations into two steps: *(i)* demonstrations acquisition and *(ii)* expert's knowledge extraction. Each demonstrator was asked to stand at the corner of an intersection, holding a smartphone at chest level, and capture, from a first-person perspective, the sequence of actions required to cross the intersection. The motivation for this particular position of the smartphone is the outcome of previous experiments carried out with visually impaired users [14, 13]. As our interpretation of the street crossing task also included an initial orientation phase to the correct direction towards the goal, demonstrators were asked to record the procedure of rotating within a range of $\pm 45^o$ about the appropriate heading from the starting corner to the goal corner.

Furthermore, as suggested by previous work [80, 81, 82], the high sensitivity of LfD techniques to the quality of demonstrations greatly impacts their generalization ability. A comprehensive set of samples $(z, a) \in D$ should capture not only the optimal behavior of the task, but also states that could only be reachable by some suboptimal action sequence.

To ensure that this was the case, the demonstrators were asked to include suboptimal behaviors in their crossings, along with the corresponding corrective actions.

Our demonstrators recorded 215 videos of approximately 25 s each from street intersections in downtown Montreal, Canada, registering the sequence of states transitioned by sighted individuals performing the task. As a compromise between data quantity and a desire to minimize redundancy of frames at a high framerate of 30 frames per seconds (fps), we extracted frames from the collected videos at a rate of 2 fps, which resulted in a total of 8125 observations.

4.1.2 Experts' Knowledge Extraction

In LfD, a transition $t \in \mathcal{T}$ between states occurs when an agent executes the actions specified by its policy. We choose to discretize the space of possible actions by dividing the agent's field of view into 12 evenly spaced vertical bins as presented in Figure 4.1, following a similar approach taken in previous research [82, 83, 84]. Each bin, $v \in \mathcal{V}, \mathcal{V} = \{v_1, v_2, ..., v_{12}\}$, is an action in $\mathcal{A}$ an expert would recommend to execute given an observed state in the street crossing task. The bins are intended to capture the heading of the goal relative to the expert's field of view, with bin v_1 corresponding to the agent having to veer maximally to the left, and bin v_{12} representing having to veer maximally to the right.

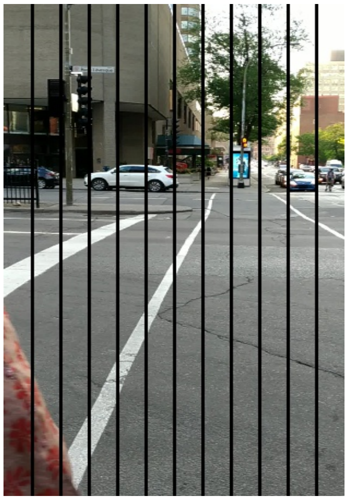


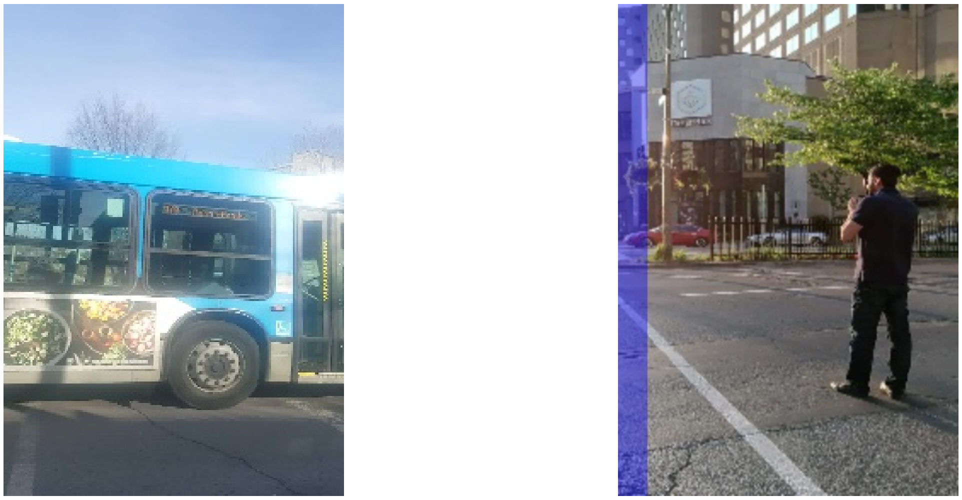
Fig. 4.1: Action space discretization into vertical bins $\mathcal{V} = \{v_1, .., v_{12}\}$ from left to right.

For situations where an expert could not identify the bin including the goal, for example, in the scenario shown in Figure 4.2a, our problem model also included an action *unknown* $\in \mathcal{A}$. As we will discuss in Section 4.2.1, this representation allowed us to experiment with different levels of granularity for the action space.

As our method did not incorporate a technique to capture the demonstrators' actions on-site, we relied on three experts' knowledge to extract optimal behavior from those observations, in a post-demonstrations procedure. For this, each expert was presented with frames randomly sampled from the observations, in a structure similar to the one depicted on Figure 4.1. They were then asked to select the bin $v \in \mathcal{V}$ that contained the position of the goal.

To ensure some resiliency to occlusions in the derived policy, we instructed the experts to choose the bin closest to the presumed goal position in scenarios in which the goal was occluded or otherwise not visible, provided that its location could be assumed (e.g., Figure 4.2b). We expected that under most conditions, a sighted individual could quickly estimate the relative orientation towards the goal from a single observation. For those exceptional cases where it was not possible to infer the target position, the experts were asked to assign *unknown* as the recommended action (e.g., Figure 4.2a).

By virtue of symmetry, we were able to mirror each image around its central vertical axis and associate the flipped image with the corresponding inverse action (i.e., swapping left-to-right with right-to-left). This allowed us to create a set of synthetic observations which, combined with the demonstration examples gathered, doubled the size of $\mathcal{D}$ and ensured a balance between the states explored and the optimal behavior observed.



(a) total occlusion of the environment (b) Optimal action inferred with goal not visible.

Fig. 4.2: Examples of demonstration frames including corner cases in our dataset.

4.1.3 Policy Derivation Technique

The literature on LfD suggests the existence of three categories of policy derivation methods: direct learning, indirect learning, and execution plans, only differentiated by how much understanding of the environment each algorithm requires while inferring a policy [76, 77]. The algorithms contained in the Direct Learning category are mostly independent of beliefs about the internal state of the environment, thus easier to implement. Then, the family of direct policy learning algorithms was our preference to solve the street crossing veering problem.

Based on the discretization of our actions space and the reduction of the observations to features, we choose to implement our policy extraction strategy as an image classification problem. A classification problem is one where a classifier $c(x) : X \rightarrow Y$ is used to predict the class y of an instance x, having $y \in Y$, $Y = \{y_1, y_2, ..., y_m\}$ a discrete set of classes. Usually, $x \in X$ is a vector $\vec{f} = \{f_1, f_2, ..f_n\}$ of features that reduce the dimensionality of the samples in X. In a supervised learning setting, the classifier is trained using a dataset $\mathcal{N}$ of samples in the form $(\vec{f_i}, y_i)$. Thus, we established the equivalence: $\mathcal{D} \equiv \mathcal{N}, \mathcal{Z} \equiv X, Y \equiv \mathcal{A}$ where the classifier $c(x) : \mathcal{Z} \rightarrow \mathcal{A}$, a CNN model, was trained to infer our policy π^* directly from samples on $\mathcal{D}$.

4.1.4 CNN for Classification Tasks

As an image usually contains irrelevant and redundant information for the resolution of visual tasks, it is better to deal with a condensed representation of such knowledge. Computer Vision techniques often rely on the extraction of salient attributes as a way to minimize the dimensionality of the information contained in an image. Manual extraction of those features requires a comprehensive understanding of the environment and the task at hand. The appearance of Convolutional Neural Networks (CNN) has come to alleviate this need while achieving human-level performance on computer-assisted visual tasks.

Notably, CNN architectures have eliminated the prerequisite of hand-crafted feature extraction algorithms by learning the required features and the task at hand, simultaneously [72]. Since ImageNet Large Scale Visual Recognition Challenge 2012 [85], CNN have obtained state of the art results [86] on benchmark datasets in image classification, segmentation or object detection like ImageNet or PASCAL Visual Object Classes Challenge (VOC) [87].

Yosinksi et al. [88] analyzed why CNN has performed remarkably well on visual tasks and concluded that the way convolutional filters are organized explains this success in part. In a CNN, each convolutional filter learns to search for specific patterns in an image. Filters on first layers of these models learn to detect low-level characteristics (e.g., edges), while filters in deeper layers are fine-tuned to compose the low-level patterns into high-level features (e.g., the shape of a flower), according to a hierarchical structure. Therefore, we used CNN architectures to convert our z component of the demonstrations $d_i = (z_i, a_i)$ to a vector $z : \vec{f} = \{f_1, f_2, ..., f_n\}$ of features and to map these features into our discrete action space $\mathcal{A}$, thus generating an optimal policy π^*.

4.1.4.1 Transfer Learning

Training a CNN for classification using randomly initialized filters, or even with traditional heuristics [89], is usually a challenging and time-consuming task as the space of the models' hyper-parameters has to be explored. Moreover, our dataset had significantly fewer instances than the ImageNet dataset (8725 vs. 1.2 million instances) and the dimensionality of the classification task is significantly lower (13 vs. 1000 classes). Consequently, the direct application of models designed for ImageNet could lead to overfitting our dataset and to the loss of generality on the predicted actions.

In this regard, the notion of transfer learning helped us to overcome those obstacles. The theory of transfer learning establishes that the knowledge on a source problem space $\mathcal{P}_s$ of a learned task $\mathcal{T}_s$ could help improve the learning of a target task $\mathcal{T}_t$ on a target problem space $\mathcal{P}_t$. How much knowledge is transferable from one domain/task to the another is directly associated with the amount of overlap between the problem areas in both [90].

Therefore, there exists a proven transferability property between features of a CNN trained on different visual tasks [61]. Although the overlapping between our demonstrations and the training samples on the ImageNet dataset is not clear, we still relied on models pre-trained on the latter as a starting point for fine-tuning different classifiers. Consequently, the high-level features of our problem were built upon the low-level features in the pre-trained models by re-training the appropriate deeper layers in each model.

Interestingly, the derivation of policies with supervised learning has presented some weakness in the past when the independence and identical distribution of the samples collected on $\mathcal{D}$ cannot be guaranteed (i.i.d principle)[82]. To guarantee such independence, each frame and the corresponding expert's action was considered a self-contained demonstration. Recent applications of LfD and CNN to navigation problems in robotics [83, 91, 84] have disregarded the sequential interpretation of a go-to-goal process thus inferring a stationary (time independent) policy. Moreover, the observations presented to the experts for labeling were randomized, ensuring their action (class) recommendation was independent of a sequential analysis of the frames.

4.2 Model Results

4.2.1 Training the Agent

The accuracy of CNN models has significantly improved in recent years relative to their computational complexity [92]. However, state of the art results remain dependent on models relying on high-performance hardware, especially Graphics Processing Units (GPUs) to carry out their inference within adequate time constraints for real-life or real-time applications. Recent work [19, 18, 93, 94] has explored CNN architectures that aim to achieve a balance between the human-level accuracy results of their predecessors and the prediction time, thus making the application of deep learning techniques to a real-time problem, such as street crossing, feasible.

In this work, we experimented with four state of the art CNN architectures. Firstly, Resnet50 [16] and Xception [17] have a reported top-5 accuracy over 90% on the ImageNet dataset. This motivated our exploration of their potential as policy extractors. Moreover, we were curious to investigate the performance of network models that have been designed specifically to achieve a balance between classification accuracy and training/inference time. Thus, we selected Squeezenet [19] and Mobilenet [18] as our testbed for a mobile deployable solution.

Our transfer learning approach was based on the fine-tuning of each model by removing the latest layers, containing high-level features, and training our custom structure from scratch. In the cases of Xception, Mobilenet and Squeezenet, after removing those high-level-feature layers from each model, we added a $3 \times 3 \times 32$ convolutional layer, followed by a $1 \times 1 \times |\mathcal{A}|$ convolutional layer, both activated with ReLUs [95]. Finally, we added a softmax activation layer with a size of $|\mathcal{A}|$. Because of the particular structure of residual networks [16], we could only add to Resnet50 an extra fully connected layer converging to the number of actions and, similarly to the models above, this layer was followed by a softmax activation layer.

After introducing these modifications, we fine-tuned the models by holding the pre-trained layers constant, and only training the final layers we added. Each model was trained with a small learning rate (0.0002), using the RMSprop optimizer [96] ($\rho = 0.9, \epsilon = 1 \times 10^{-8}, \delta = 0.0$) and a categorical cross-entropy loss. The values of these hyper-parameters were selected empirically. With this configuration, we aimed to ensure the stability of the pre-trained values of each model.

We then experimented with reducing the dimensionality of the action space. Starting from the arrangement of 12 bins, we generated the following three configurations:

- 4-actions space: $\mathcal{V}_1$, by combining $\{v_1, ..., v_4\}$, $\{v_5, ..., v_8\}$ and $\{v_9, ..., v_{12}\}$ into $\{v_{\text{left}}, v_{\text{straight}}, v_{\text{right}}\}$ respectively, plus the *unknown* action, as shown in 4.3a.

- 8-actions space: $\mathcal{V}_2$, by combining $\{v_2, v_3\}$, $\{v_4, v_5\}$,..., $\{v_{10}, v_{11}\}$, reserving bins $\{v_1\}$ and $\{v_{12}\}$ for those situations when the goal is not visible but its position can be inferred, as shown in Figure 4.3b.

- 13-actions space: $\mathcal{V}_3$, retaining the full configuration of as shown in Figure 4.3c.

For each of these configurations, we modified the associated Softmax layer to accord with the sizes of $\mathcal{A}_1 = \mathcal{V}_1, \mathcal{A}_2 = \mathcal{V}_2, \mathcal{A}_3 = \mathcal{V}_3$, and added $v_0 = \textit{unknown}$. We then trained the CNN classifiers and evaluated their performance.

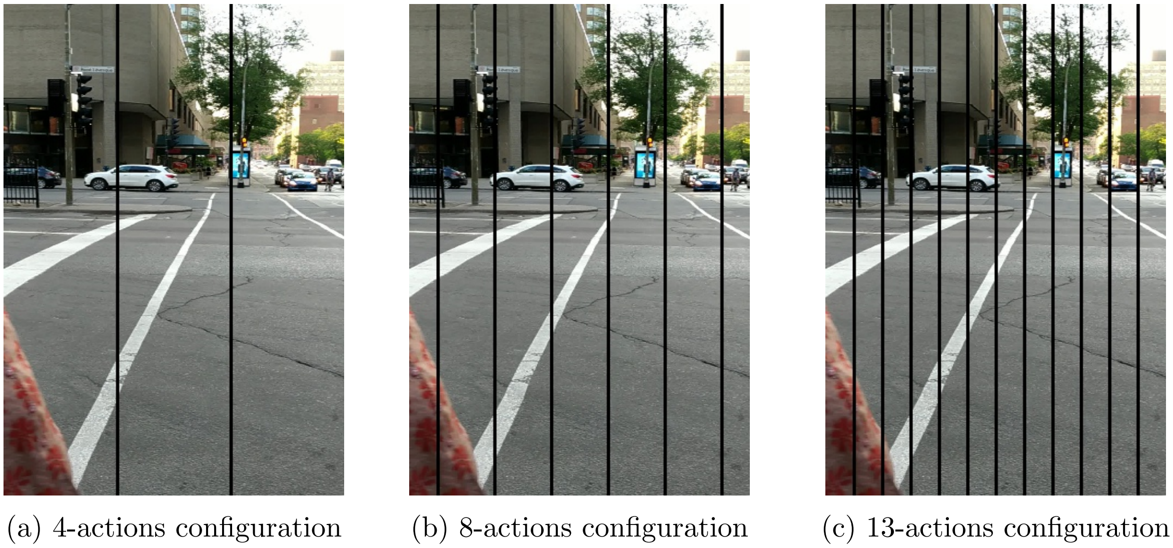

(a) 4-actions configuration (b) 8-actions configuration (c) 13-actions configuration

Fig. 4.3: All actions-space configurations experimented while training and testing the agent.

4.2.2 Testing the Agent

To evaluate the generalization of the learned policy, we created a second demonstration dataset from different intersections that were not included in the training set. Following the procedures described in Section 4.1.1, a supplementary collection of 51 videos was acquired, resulting in a new set $\mathcal{O} : \mathcal{Z}_o \times \mathcal{A}_o$ of 1170 observations. The optimal action for those samples was crowd-sourced to another ten experts who labeled each sample at least five times. The conditions described for labeling the initial set $\mathcal{D}$ were also followed here.

Table 4.1 presents the accuracy of the derived policy, applied over the observations on $\mathcal{O}$. These results are computed based on the best-predicted action of the classifier compared to the action that received the most votes from our experts. However, we note that best-action accuracy metrics are not meaningfully indicative of the model's actual performance on a practical task. Instead, Table 4.2 presents the mean absolute error in the agent's predicted action, a measurement computed by taking the absolute difference between the index of the action inferred by the policy from an *observation* and the index of the winning vote from the experts. Considering that our distribution of the action space is dependent on the spatial arrangement of the bins, we excluded the results of the action *unknown* in this calculation.

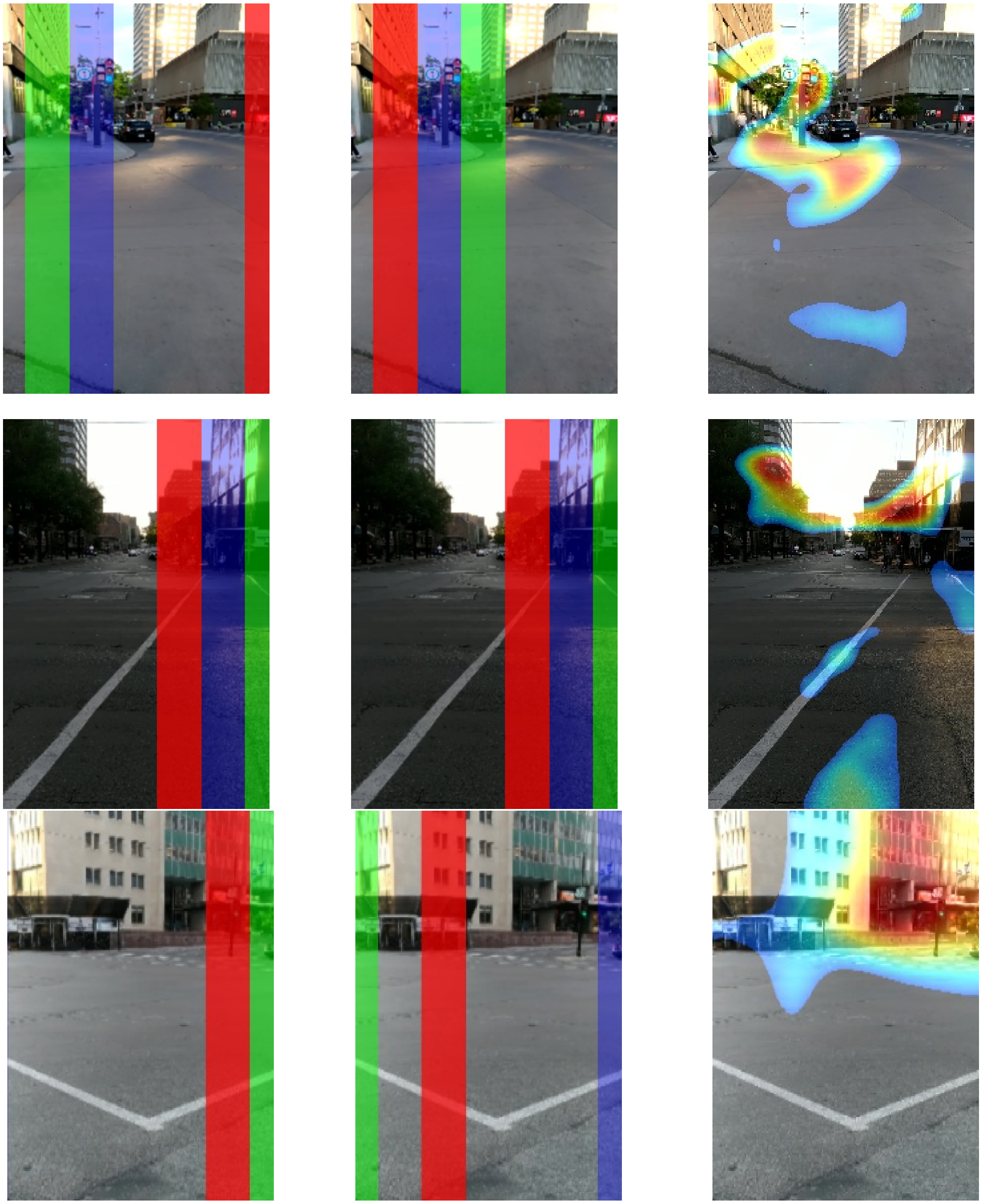
Model	4-Action	8-Action	13-Action
ResNet-50	0.746	0.635	0.503
Xception	0.822	0.615	0.526
Squeezenet	0.775	0.483	0.393
Mobilenet	0.822	0.599	0.467

Table 4.1: Accuracy of each model in predicting the correct action, compared to the experts' optimal action.

Model	4-Action	8-Action	13-Action
ResNet-50	0.27 ± 0.03	0.61 ± 0.07	1.14 ± 0.12
Xception	0.20 ± 0.03	0.59 ± 0.07	1.05 ± 0.12
Squeezenet	0.26 ± 0.03	0.83 ± 0.07	1.37 ± 0.12
Mobilenet	0.20 ± 0.03	0.71 ± 0.08	1.24 ± 0.12

Table 4.2: Each model's mean absolute difference between predicted action and the experts' optimal action, presented with the corresponding 95% confidence margin.

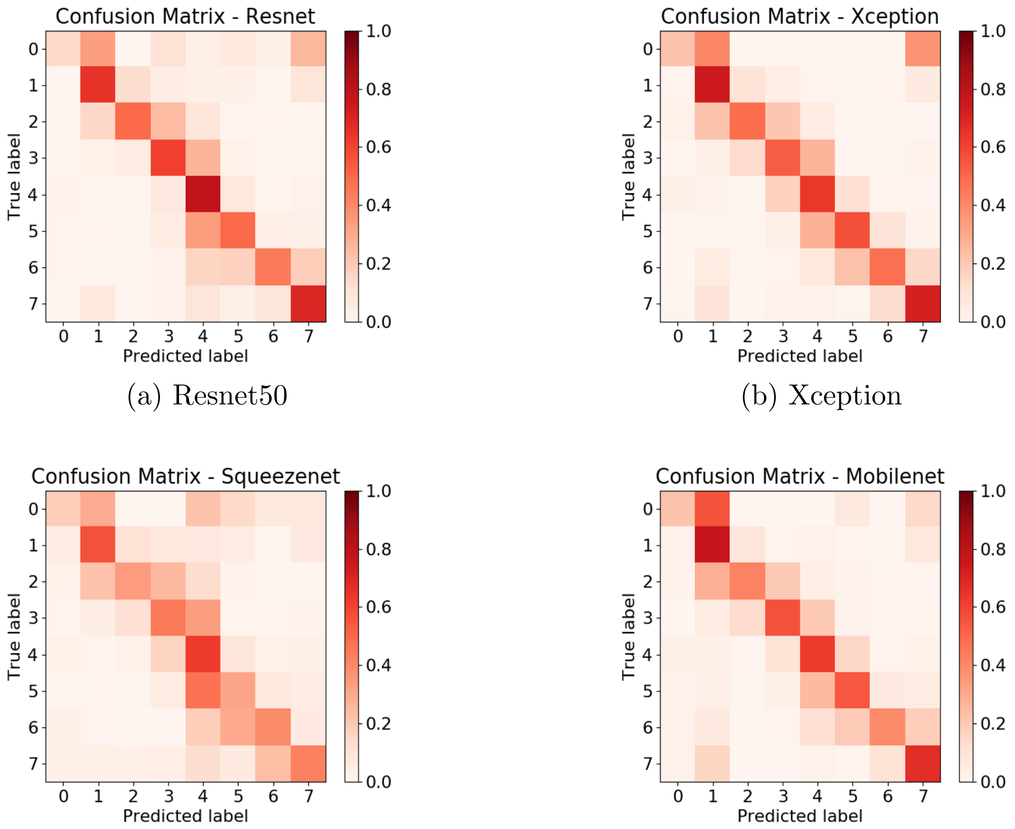
As can be seen, relying solely on the accuracy metric would suggest that the agent exhibits poor performance. However, given the mean absolute error reported—typically within a difference of a single bin—the average performance of the system is actually satisfactory across all model types and action space configurations. This can be verified by analysis of the confusion matrix for each action-space configuration. One can observe in Figure 4.5 a strong tendency around the diagonal in all four models, indicating that errors in the agent's prediction are most often the result of confusion with an adjacent, i.e., very similar, action. Thus, a mean absolute error metric is more appropriate than a simple correctness percentage score to characterize the performance of the model. Although we only present here the 8-actions configuration, similar behavior was exhibited for the other action-spaces tested.



(a) experts' prediction (b) agent's prediction (c) activation map

Fig. 4.4: MobileNet top-3 predictions ($blue, green, red$) vs. experts' predictions on the 8-action-space. A missing bin corresponds to *unknown*. (c) shows the CNN activation maps [3].

It is also interesting to note that for situations where the experts' optimal action was *unknown* (i.e., the correct label is 0), the agent would most often confuse it with the extreme veering conditions (i.e., actions 1 and 7). This suggests that when the expert is unsure of the required action, the agent's predictions is to preform a large rotation. We suspect that this behavior is related to the way experts chose the optimal action in the training demonstrations; when the goal was not seen, the expert would choose the edge column that they guessed was the best direction to which one should rotate. It is evident from the last row of Figure 4.4 and Figure 4.5 that the agent has also learned this behavior. Although some perfect agreements between the policy and the expert's judgment are represented in the first row of Figure 4.4, there are still scenarios in which the goal is occluded and the policy is not capable of inferring the correct behavior, as shown in the third row of Figure 4.4.



(a) Resnet50

(b) Xception

(c) SqueezeNet

(d) MobileNet

Fig. 4.5: Confusion Matrix for each model trained on the 8-action-space configuration.

4.3 Mobile Application Design

4.3.1 Model Selection

For spatial tasks, performance evaluation of the resulting models based on the mean absolute error metric is more meaningful than using the accuracy of the best action selection [10]. While all the models we tested exhibited consistent performance on the prediction of the optimal action, they differed significantly in computational cost. Since we were focused on deployment in a real-time mobile environment on a resource-constrained Android platform, we considered factors of inference time, memory, off-line storage requirements, and battery consumption.

We also examined the impact of action-space configuration when mapped to an auditory feedback method. While the 4-actions (3 actions plus the *unknown*) arrangement achieved the best prediction results, it did not produce sufficient spatial information to present continuous feedback across the different possible veering conditions. We ruled out use of the 13-actions discretization due to its overly small level of granularity. This left the 8-actions (7 spatial actions plus the *unknown* action) discretization as the best choice for the experiments we conducted.

Table 4.3 shows the measurements[1] obtained for each of the CNN architectures on our test device (a Samsung Galaxy Note 5), which exhibited the best performance across the sample set of smartphones available. Our target is to obtain at least two predictions per meter of travel. Considering that an average sighted individual walks at a speed of 1.4 m/s and that a blind pedestrian tends to walk at a slightly slower rate [97], inference times above 500 ms fall at the limit of what we deem acceptable. For this reason, we considered Xception and Resnet50 models inadequate for the real-time mobile context (1256 and 1052 ms per prediction, respectively), even if their mean absolute error was the best among all trained architectures. On the other hand, MobileNet and Squeezenet were sufficiently fast (309 and 163 ms per prediction, respectively) that we could update the optimal action at a higher rate. Thus, we based our experiment setup on MobileNet, ignoring the lower-performing Squeezenet, with the aim of balancing between the inference time and mean absolute error of the predictions. As more computationally powerful devices become the prevailing norm in the smartphone market, we expect these models might become feasible.

[1] All models were quantized, when applicable, before deployment. See https://www.tensorflow.org/performance/quantization

Criteria	Resnet50	Xception	MobileNet	Squezeenet
Inference time (ms)	1052 ± 160	1256 ± 150	309 ± 40	163 ± 12
CPU load (%)	41.13	43.47	31.43	20.13
Power consumption (mAh)	4166	3941	5294	2980
Memory footprint (MB)	116.18	103.85	26.17	16.23
In-disk size (MB)	90	82	14	3

Table 4.3: Mean inference time (milliseconds) and standard deviation, mean CPU load (percentage), battery consumption (milliampere hour), memory footprint (megabytes) and in-disk size (megabytes) factors across the trained CNN models, evaluated using our prototype application deployed on a Samsung Galaxy Note 5.

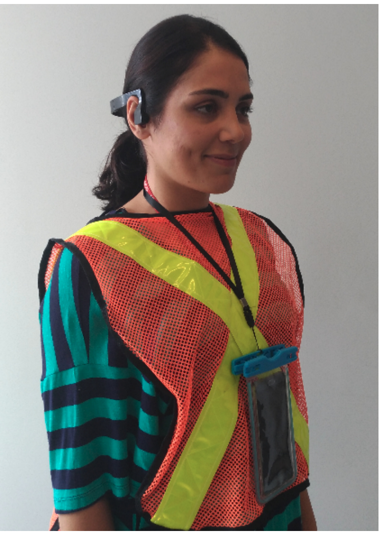


Fig. 4.6: Users' setting carrying the smartphone in a lanyard worn around the neck.

Despite the accuracy observed in these models in a static-frame setting, real-world testing conditions were less forgiving. Since our test users carried the phone in a lanyard worn around their neck (Figure 4.6), the camera view constantly oscillated as a result of body sway while walking. This is shown in Figure 4.7 where a tester crossed an intersection with this setup, and the heading angle (in degrees) was recorded. This would negatively impact the results for both of the evaluated feedback methods, as the location of the goal was similarly oscillating during the crossing.

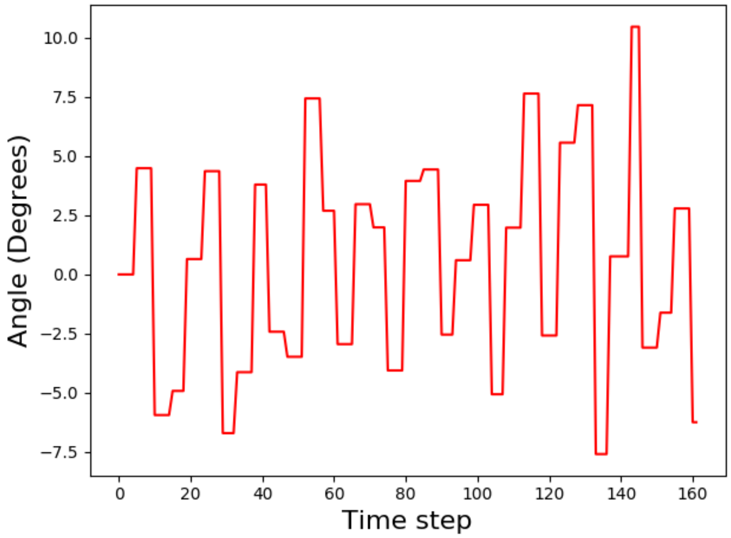


Fig. 4.7: The heading angle (in degrees) of a tester crossing an intersection while carrying the smartphone in a lanyard worn around the neck, where the angle was recorded using Android API's *TYPE_GAME_ROTATION_VECTOR* [4] (this sensor fusion is further explained in Section 4.5). Since the tester was walking a straight path, ideally the angle would be in the proximity of 0 degrees.

To mitigate these effects, we implemented a weighted voting scheme (Equation 4.1) that updates its output as a function of previous optimal actions predicted by the CNN architecture, giving more weight to recent predictions. This resulted in the rendering of a more stable output signal, as tested on the authors.

$$\begin{aligned} V_i &:= \gamma \cdot V_i, \qquad \forall i : 1..n \\ V_{action} &:= V_{action} + 1.0 \\ \text{output} &= arg \max_i (V_i) \end{aligned} \tag{4.1}$$

Where V is a vector holding the current voting power of each action and $0 < \gamma < 1$ represents how quickly we want to forget the previous voting powers. V_{action} is the voting power of the newest predicted action provided by the model. The action with the highest voting power is selected as the optimal action to render.

4.3.2 Feedback Modality

Many rendering methods have been explored in previous work, including both auditory and haptic signals. As representative examples, Panëels et al. [13] designed a constant auditory signal whose balance between the two ears indicated the direction of veering, and Ross et al. [53] delivered haptic feedback with an array of body-worn actuators. However, a constant auditory stimulus would distract the user from focusing on the natural sounds of the environment, and a haptic array remains impractical for widespread deployment.

Instead, we compared two techniques that had been employed successfully by previous systems. The first renders a "warning-style" tone on the side of the direction of veering, but remains silent when the user is not deviating [13]. The second produces spatialized audio to render the location of the goal as a virtual beacon [53] similar to that of an actual APS system [98]. The stimulus used is the same as the APS sound, presented every 1200 ms. Hereafter, we will refer to this method as the "beaconing-style" method. Both types of stimuli were produced using an open source implementation of the OpenAL platform API for Android-based systems.

4.4 Evaluation with Visually Impaired People

The evaluation of the system consisted of a two-step process. First, a pilot study was conducted with sighted participants in order to compare the two auditory feedback designs described above. The application was then updated with the preferred feedback design. The full experiment, conducted with fully blind participants, evaluated the performance of the system in helping people with visual impairments align themselves with the crossing and remain within the limits of the crosswalk while crossing.

4.4.1 Intersection and Environment

The same test intersection was used for both the pilot study and the full experiment. This intersection was chosen with the criteria of proximity to the university to ensure a short experiment, and a low-to-moderate amount of traffic flow, thus, minimizing stress on both the participant and experimenter. Each crossing had unfaded two-line pedestrian markings, thus facilitating determination of veering of the participant outside of the crossing lanes. The longer crossing was 11 meters wide with two vehicle lanes while the shorter one was

8 meters wide with a single vehicle lane. The longer crossing also included a bicycle path, providing another factor that may influence misalignment from hearing a bicycle. Both roads had one-way traffic flow, which may pose an additional challenge to participants who try to align themselves using this cue.

4.4.2 Task Procedure

The same process was followed for each crossing in both the pilot study and the full experiment. Each trial consisted of two phases: initially aligning with a given crossing and then crossing the intersection. To reduce the impact of learning on the results, each participant first went through both phases at a training intersection (different from the test intersection) to familiarize themselves with the auditory feedback provided by the application.Then, after proceeding to the test intersection, they carried out a series of crossing trials under the two test conditions. For the pilot, these conditions were the two auditory feedback approaches, whereas for the full experiment, with blind subjects, we tested with and without the auditory feedback active.

Each trial followed the same procedure. Participants were kept 3 meters away from the first corner. The experimenter then guided the participant to the corner of the intersection, ensuring that their initial heading deviated approximately 30° from the correct orientation, i.e., the center of the intersection. The experimenter then gave the participant a hint as to the location of the goal, e.g., “Your goal is towards your left”.

At this point, the experimenter instructed the participant to orient towards the goal without taking steps in any direction, ensuring the participant remained in a safe state at the corner. The experimenter then took three steps away and waited for the participant to indicate when they thought they were properly oriented. If the participant was judged to be misaligned, the experimenter would orient them in the correct direction. Once the traffic lights indicated that it was the pedestrians’ turn to cross, the experimenter confirmed that it was safe, and accompanied the participant throughout the entirety of the crossing, remaining a few steps away to ensure that the participant did not use the experimenter for guidance. If the participant happened to veer outside of the crossing lanes, the experimenter halted the participant, moved them sideways to the center of the crossing, corrected their heading towards the goal, and instructed the participant to continue.

Each trial was recorded with a video camera by a second experimenter who remained at the starting corner, observing the crossing behaviour of the participant from behind. In addition, user comments following each trial were written by the experimenter on a paper pad. At the end of the experiment, for the pilot and the full study, participants were asked to complete a Likert-scale questionnaire that assessed their comfort with the auditory feedback, their level of annoyance with the system, and the confidence they had in the application.

4.4.3 Pilot Study: Choosing the Better Audio Feedback Design

To choose the better feedback method, we conducted a pilot study with sighted participants, who were asked to keep their eyes closed during each trial. Initially, we intended to blindfold the participants but decided against it for safety reasons, since we can warn participants to open their eyes at any time should a dangerous situation arise. During the pilot, participants first trained with their eyes open to learn how much they would need to pivot during the initial alignment. At the test intersection, before approaching the corner, the experimenter asked the participant to close their eyes and to try to keep them closed unless instructed otherwise, or if they felt uneasy. At the end of each crossing, the participants was allowed to open their eyes while providing their comments.

Figure 4.8 shows the crossing pattern followed throughout the pilot study. All users began at origin O_1 crossing in the upward direction then returned to O_1, before crossing in the left direction. This was done twice, once with each feedback method. Once participants had completed both crossings starting from O_1, we proceeded to O_2. The initial auditory feedback condition was balanced across participants, i.e., participants who started with the beaconing-style guidance at O_1 started with the warning-style feedback at O_2 and vice versa.

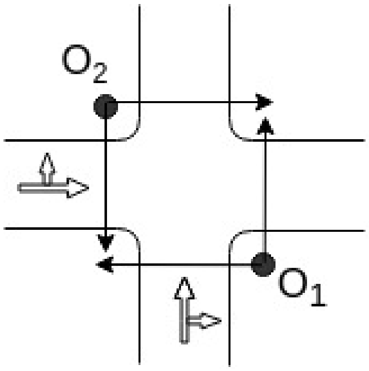


Fig. 4.8: Crossing pattern followed during the pilot study with sighted participants.

For the pilot experiment, six sighted volunteers (4F/2M), ages 21-28, were recruited from our research laboratory at the university. Qualitative results from the pilot study demonstrated the superiority of the warning-style feedback. Four of the six participants expressed a preference for this method. Although none of the participants reported any hearing impairments in the pre-test questionnaire, three of them demonstrated a great deal of difficulty with the beaconing-style guidance. One participant, whose first trial was conducted with this feedback method, expressed anxiety during the first trial, and subsequently commented, “This is really scary. I can’t tell if the sound is centered or if I should go towards the right.” Another participant expressed how much she preferred the warning-style feedback, saying, “The goal of keeping the system quiet is a much easier one to follow in a high risk environment such as crossing an intersection”. Another surprising finding from the pilot was the degree of confidence gained by participants while crossing intersections with their eyes closed. One participant expressed this in the post-test questionnaire, stating, “If someone that is used to seeing feels very safe after a few rounds of crossings, I would assume a blind person would be very comfortable with it”. In addition, we observed an apparent higher cognitive load under the beaconing-style guidance condition. Some participants when crossing under this condition would take one step forward, then stop and listen to the feedback. This is clearly undesirable when crossing intersections as we would not want users to stop and think after every step.

Panëels et al. [13] explored the differences between two different beaconing-style feedbacks with sighted individuals. The authors also conducted the pilot tests with two blind participants. After these pilot tests, the blind participants informed the authors that a beaconing-style feedback goes against what they are thought during orientation and mobility training. Combining this result with our observations from the pilot study, the warning-style feedback was chosen for the evaluation of the application's performance with blind participants.

4.4.4 Full Study: Evaluating Performance with Blind Participants

To carry out ecologically valid assessment of the effectiveness of our system, the experiments were conducted with blind participants. The intention was to replicate what a blind individual would encounter when exploring new environments, or at intersections in a known environment without a large amount of traffic flow, both scenarios having been identified as challenging for people with visual impairments.We began with the following hypotheses about the potential impact of our system:

- **H1:** Before crossing, users are more likely to be correctly aligned with the opposite corner using the application than they would be without it.
- **H2:** Using the application, users take a shorter amount of time to initially align themselves with the opposite corner.
- **H3:** The application reduces the likelihood that a visually impaired user will veer outside the crosswalk.
- **H4:** Using the application does not increase the total amount of time a user takes while crossing.

4.4.4.1 Methodology

Figure 4.9 shows the crossing pattern followed throughout the experiment. As shown, participants were given the task of crossing all four sides of the intersection starting from the origin O_1 moving in a counter-clockwise loop. At each intersection, an image was captured on the experiment phone when the participant indicated they felt properly oriented, capturing the orientation of the participant relative to the target corner. This cycle was repeated four times, twice using the application (condition A) and twice relying only on their mobility training techniques (condition B), for a total of $4 \times 4 = 16$ trials.

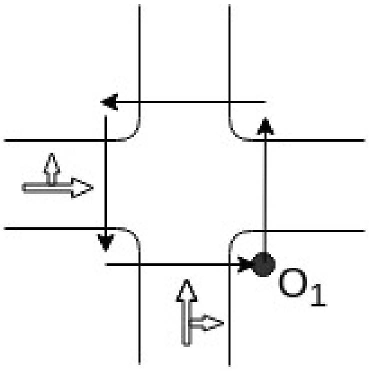


Fig. 4.9: Crossing pattern followed during the full experiment with blind participants.

For this experiment, ten blind participants (5F/5M), ages 26-58, were recruited from [Institute name omitted for blind review] and compensated with $20 for their time plus transportation cost. All participants were fully blind, mostly from birth. One participant (F) had to leave early and did not complete the final trial under condition B, so their data was excluded from the analysis. Another participant (M) who exhibited a visible walking problem was unable to complete the experiment, and was therefore also excluded from the analysis. Six of the remaining participants used a white cane during the experiment and two used a guide dog. Four of the participants followed an ABBA circuit, i.e., one full cycle with the application (A) followed by two full cycles without the application (BB) and one last cycle with the application (A). The four other participants followed the opposite (BAAB) cycle.

4.4.4.2 Results

We start by testing the first hypothesis (H1). Four raters were tasked with determining whether the participant was correctly aligned, based on the images recorded prior to each crossing. These raters, one of which is an author of this work, were presented with each image in random order, without knowledge of the experimental condition, and asked to assign a binary label (aligned/not aligned) to each image. The raters had a Fleiss-Kappa value of 0.810, which represents an almost perfect level of inter-rater agreement. Figure 4.10 shows the results for the eight participants. As an example, P5 was misaligned four out of eight trials under the control condition and twice misaligned under the auditory feedback condition. Running a paired t-test showed a statistically significant ($p = 0.0103$) difference between the two conditions, supporting our hypothesis that the application reduces the likelihood that a blind individual will initially be incorrectly aligned. In addition, we found an effect size of 1.56 indicating a very large effect between our two conditions.

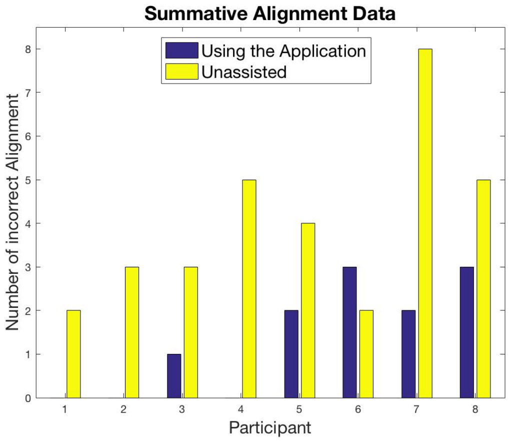


Fig. 4.10: Comparison of the number of misalignments with and without the application for each participant.

To test our hypotheses H2, H3, and H4, we analyzed the video recordings from each trial. We extracted the time each participant took to initially orient themselves, starting from the moment the experimenter backed away, the number of times participants veered outside the crosswalk during the crossing phase, and the time to cross the intersection, starting from the first step. Unfortunately, due to an error in the recordings, data from one participant's data (F) was missing some of the parts required to extract the useful information to assess H2, H3 and H4. This unexpected situation left us with only seven participants to test the three remaining hypotheses.

We proceeded to test our second hypothesis, H2. As we noted in our observation sessions with visually impaired individuals, the orientation time does not depend on the current corner but rather on the amount of traffic flow at the intersection. This is particularly the case for intersections in unknown neighborhoods, such as the one used in our experiments, allowing us to compare all orientation trials with and without the system, regardless of which crossing corner the participant was located. Since we avoided rush hour traffic during the timing of the experiments, we can reasonably assume that traffic flow was reasonably similar across participants.

During the experiments, we perceived a large variance in each participant's first and second trial under the auditory feedback condition. For example, one participant took 237 seconds to orient himself in the first trial while, in the second trial, he only took 21 seconds. As such, we run a paired t-test to test the assumption that there is a significant learning effect between the first and second trial with the system. As a result, we obtained a p-value of ($p = 0.014$) which supports this conjecture. To test H2, we conducted another paired t-test comparing only the second trials of using the application versus the control condition. This was done to avoid biases caused by learning effects from first use with the system. The results are statistically significant ($p = 0.0269$) allowing us to conclude that, following an initial training session with the application, participants can align themselves faster with the opposite corner.

Next, we test our third hypothesis, H3: The application reduces the likelihood that a visually impaired user will veer outside the crosswalk. To determine whether participants will be less likely to veer, we extracted the number of times each participant deviated outside the crosswalk under each condition. Figure 4.11 displays the number of times each participant veered outside the crosswalk with and without the application. For example, P2 veered outside the crosswalk in three out of eight trials under the control condition, and veered once out of eight trials under the auditory feedback condition. There is only one participant (P3) whose data show a negative outcome while using the application. We note that this participant used a guide dog during the test and, following the trial, she reported that even though the application was telling her to correct, she ignored it since she did not want to correct the dog's behavior. Although we cannot yet reject the null hypothesis that the system reduces the likelihood of veering (paired t-test $p = 0.0519$), we observe a positive trend towards the validation of this position. Observing the results in Figure 4.11, it would appear to support our hypothesis, with only one participant exhibiting a negative response when using the application versus the control condition. In addition, we found an effect size of 0.98 indicating a large effect between our two conditions. With further testing on more blind participants and addressing some of the apparent shortcomings of the current application (as detailed in the Section 4.5) we anticipate it will show statistical significance.

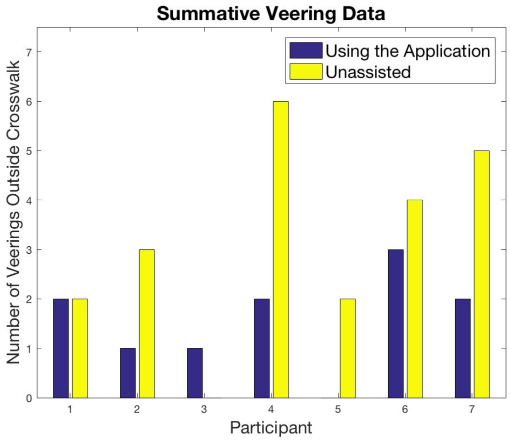


Fig. 4.11: Comparison of the number of veerings with and without the application for each participant.

In testing the last hypothesis (H4), we average each participant's two trials under each condition. We then conduct four paired t-tests evaluating the difference in crossing times between each crossing under both experimental conditions and found no significant difference in crossing times (Intersection 1: $p = 0.7833$, Intersection 2: $p = 0.1340$, Intersection 3: $p = 0.1755$, Intersection 4: $p = 0.1106$). This outcome supports our hypothesis that using the application does not increase the total time it takes to cross a given intersection.

4.5 Iterative Improvements of the Application

The qualitative and quantitative analysis of the collected data pointed to several areas where the user experience could be improved or the perfomance of the application was lacking.

First, having the smartphone held by a lanyard around the neck did not take into account the variations in abdomen size of the participants. This significantly affected the viewpoint of the camera in a few cases (example shown in Figure 4.12), resulting in a completely transformed perspective of the image frames fed into the model, relative to those employed during training. Examination of the qualitative data obtained from affected participants indicates that this issue heavily impacted prediction, resulting in poor accuracy of the feedback. Another major drawback of this configuration is the camera sway caused by human walking. Although we addressed this problem in part through Equation 4.1, our use of a weighted average over the last five predictions is of questionable robustness. Improved solutions could range from a video stabilization procedure, to some form of frame filtering, taking advantage of the built-in sensors of the smartphones to determine the appropriate instant to acquire a frame.

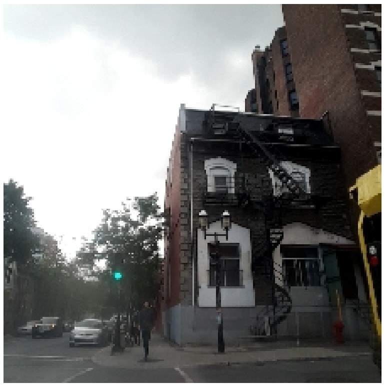


Fig. 4.12: Distorted image perspective due to the user's anatomical characteristics.

Although Panëls et al. [14] reported that the phone in the lanyard configuration is the preferred one by blind participants, this may not be true when dealing with camera-based applications. Furthermore, the qualitative data collected indicates a few blind individuals may opt out of wearing their smartphone in this fashion due to security concerns. Consequently, future studies should resolve the appropriateness of smartphone-only systems or if a mixed wearable-smartphone solution could help improve blind users' experience to a less constrained environment. In order to further improve the application from a robustness perspective, we solve the problem by employing a torso-strapped solution that holds the mobile phone as shown in Figure 4.13. We acknowledge that this solution may not be preferred by users, however we choose to focus on other aspects of the application and leave this issue for future work.

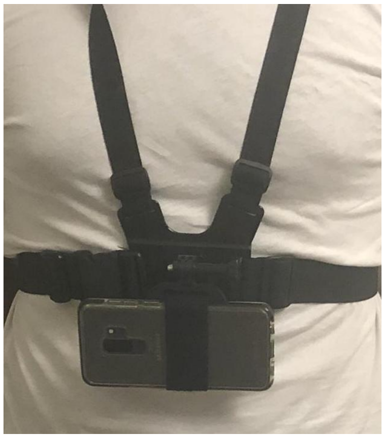


Fig. 4.13: New body harness to hold the smartphone for future experiments.

Secondly, further opportunities for improvement were apparent in the design of our feedback, which rendered a continuous sound on the side of the veering and no sound when the users' alignment with the opposite corner was accurate. The 8-action model, which divided the camera image with a field of view of 77° into seven directional bins, proved to be excessively pedantic regarding what constituted correct alignment. Only the central bin was mapped to the "silent zone", and thus, corrective feedback was provided overly frequently, alternating between the left and right channels, even when the users' orientation was reasonably well aligned with the opposite corner. We found that this uncompromising feedback was necessary in determining the correct location of the opposite corner when the users were initially aligning themselves. However, once the correct heading is determined, there is no need for the application to be so meticulous.

This effect can be mitigated by applying a Kalman filter [78] on the predicted bin. This was done as a replacement to the current historical averaging algorithm described in Equation 4.1. Since we are in a one-dimensional space in this work, the vectors and matrices in Equations 3.10-3.14 are all scalar values. In our application, the state is the orientation of the phone represented by the action values from 1 to 7. The state transition value F, which represents the model of the phone in the world, is assumed to be 1 since we do not have a model of how the phone might rotate, and we assume little rotation between predictions. We omit the action u in this work as we have no means of modelling how the user will rotate based on the feedback she obtains. Since our state and our measurement are in the same numerical space, H is assigned a value of 1. As we noted in Section 3.3, it is difficult to estimate the process uncertainty Q. We therefore assign it a value of 0.01, determined through empirical testing. The value of the error in our obtained measurements (R) is assigned a value of 0.8, which is the mean absolute error of the model. We initialize the state of the phone as properly oriented, i.e., $x = 4$, and the uncertainty of the phone's orientation as having an uncertainty equal to the mean absolute error of the model, i.e., $P = 0.8$.

Figure 4.14 shows a graph comparing simulated raw predictions from the model compared to filtered predictions from the weighted average approach (from Equations 4.1) and from the application of the Kalman filter. As we can see, the most stable signal is the one filtered through the Kalman filter, with the predictions obtained from the weighted average approach having responses to the noise of the raw output, such as between 40^{th} and 110^{th} time-steps.

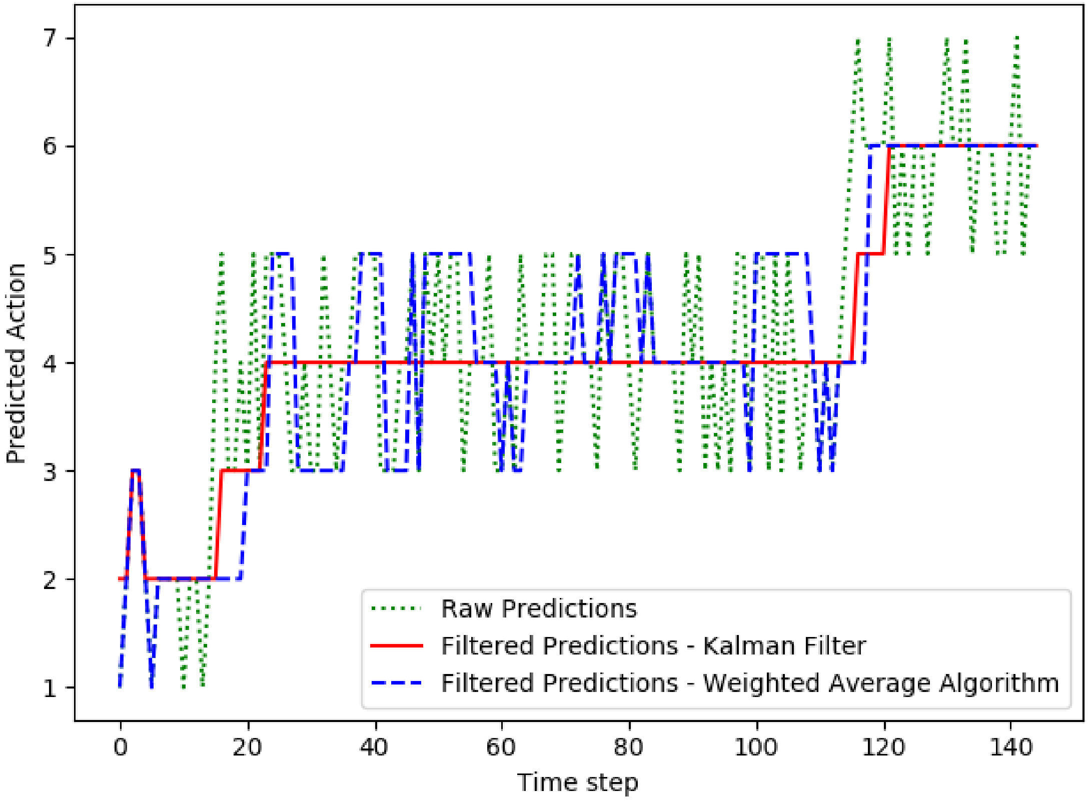


Fig. 4.14: A comparison between the simulated raw predictions from the CNN model compared to filtered predictions using the weighted averaging algorithm from Equations 4.1, and the application of the Kalman filter.

Thirdly, solely relying on the model is perhaps not the ideal way to provide heading feedback. As we mentioned earlier in this chapter, the model is not always accurate and the oscillations in the feedback, which occur at times even when users are completely stationary, can be problematic. In addition, as we can see from Table 4.3, the battery consumption is relatively high, even when using the agent with the MobileNet CNN for policy extraction. Using this CNN results in the feedback to the user being delayed by 300 milliseconds (inference time in Table 4.3), which would be acceptable, provided the model was generalizing well. However, correcting the model's inaccuracies by means of historical averaging (Equation 4.1) or Kalman filtering makes the application too slow in providing feedback while walking. To mitigate these effects, we can build on previous work [13, 33, 53] by using the built-in IMU sensors now available in commercial smartphones to provide directional feedback. As noted by Panëels et al. [13], the compass can be influenced by magnetic interference from the environment, e.g., cars or larger vehicles, which can cause offsets of over 30 degrees. Fortunately, the Android API provides access to various fusions of IMU sensors. In particular, in this work we use the *TYPE_GAME_ROTATION_VECTOR* [4], which does not depend on the magnetometer, and, according to the documentation [4], if the phone is rotated and returned to the same real-world orientation, it should report the same rotation vector. The data provided by *TYPE_GAME_ROTATION_VECTOR* fuses the built-in gyroscope and accelerometer to provide stable and accurate readings regarding the phone's rotation. We refer the reader to the documention for more information on the specifics of the sensor fusion algorithm [4]. For this work, we do not require knowledge of the direction to the magnetic north since we can use the model to inform us of the correct heading angle. With the use of IMU data combined with the model, one can use different schemes to provide the user with heading feedback. In this work, we choose to have the model be the initializer for the IMU data. This can be explained by the following three steps:

1. The model determines the correct heading by obtaining ten consecutive predictions of the 4^{th} (central) bin. This threshold was determined empirically.

2. The IMU sensors are started, taking the first angle as the reference angle, which was determined by the model as the correct heading.

3. From thereon, the model ceases to provide predictions, and only the sensor readings are used for heading feedback.

Finally, we note that the model, in its current state, is not ready for full deployment as we cannot guarantee its performance. This was apparent primarily during the crossing phase where much of the feedback provided was oscillatory. We are confident that the new harness shown in Figure 4.13 would alleviate some of these problems, but would not completely eliminate them. We hypothesize that improvements in the architecture and the collection of more data should improve its performance. However, first we wanted to observe the marginal effects of switching to the new harness and to the use of IMU data.

After updating our method of harnessing the smartphone to the user, and switching to the use of IMU data following the established correct heading by the model, a series of experiments were performed with one of our blind colleagues who has been an enthusiastic test user of our system since the outset of this project. This participant is experienced with beta testing mobile applications for the visually impaired community, and a frequent outdoor traveller of various areas in downtown Montreal (Canada). In the following subsections, we will go through the three iterative rounds of improvements of the application, presenting the feedback we received and reflecting on our observations. The assessment we present here is purely qualitative, with the knowledge that a quantitative assessment of the application will be required before it can be deployed. We leave quantitative assessment of the application for future work.

4.5.1 First Round of Improvements

We met with the participant near the Loyola campus of Concordia University in Montreal, and proceeded to testing the updated application at six intersections. The following describes a series of observations and enhancements made in response:

- Before testing the application with the user, the experimenter explained what the auditory feedback meant, i.e., that feedback is rendered on the same side of veering, that it will be quiet if they are walking in the correct direction and also demonstrated what the unknown sound is. Instead of having to say this before every experiment, we integrated a tutorial using Android's text-to-speech engine to inform the user of the feedback every time the application is started. This tutorial plays as follows:
    - "Rotate away from the auditory cues as slowly as possible. If no sound is produced, it means you are correctly oriented."
    - $Left\_Ear\_Cue$ "indicates you should rotate right."
    - $Right\_ear\_Cue$ "indicates you should rotate left."
    - $Unknown\_Cue$ "indicates something is obstructing the view."
- Since we do not want the user to begin crossing unless the correct heading was determined by the model and the application switched to using IMU data to provide the feedback, it would be informative to have a speech feedback that indicates that the correct heading was found. This was determined to be critical as the experimenter would constantly have to inform the user when the correct heading was determined.

To solve this, we use the text-to-speech engine provided by the Android API [4] which renders the following at the start of the application: "Please begin slowly rotating while I determine the correct heading", and renders the following when the correct heading is determined: "Correct heading determined. When you know it is safe, begin crossing".

- Observing the user while using the application, it was apparent that in the initial orientation phase, he would try to rotate quickly to locate the opposite corner. However, this causes problems in determining the initial heading direction as it can result in the model overshooting the prediction from bin 1 to bin 7, making it difficult for the user to stabilize the prediction. Reflecting on this finding, we recall that from the full experiment described in Section 4.4.4, this phenomenon was observed with several of the participants where the experimenter was forced to repeatedly tell the participant to rotate slower. To solve this problem, we implemented a method that obtains the angular speed of the phone, and therefore the angular speed of the user, using the built-in gyroscope calibrated using the Android API [4]. If the gyroscope reads an angular speed of $30^o/s$, using the text-to-speech engine it reads "rotate slower". This was implemented for the entire crossing, i.e., in the initial orientation phase which uses the model, and the subsequent crossing phase which uses IMU data for feedback.

4.5.2 Second Round of Improvements

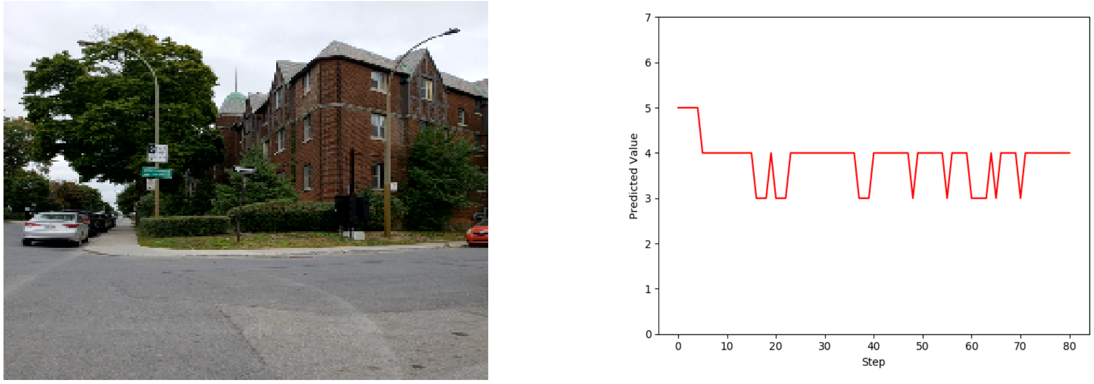
With the adjustment from the first round, we met with the participant again and experimented with crossing three intersections near the Loyola Campus of Concordia University. We present the feedback and observations we obtained in the following:

- A problem with the gyroscope is that it can have random peaks in its readings during walking. This can be due to the compensation of the gyroscope performed by the Android API, which uses the accelerometer. And since the accelerometer readings are affected by sudden impacts, such as every step, this has an effect on the final gyroscope readings. This resulted in the frequent rendering of the "rotate slower" feedback method while crossing, causing some confusion to the participant. We could try to correct this by choosing when the readings should be obtained and assessed, e.g., acquiring the readings shortly after we know the user has taken a step to avoid the sudden spikes in accelerometer readings. However, we hypothesize that this is not necessary during the crossing phase since the users should naturally be walking without too many abrupt rotations. In addition, slow rotations are mainly necessary when the model is determining the correct heading, and not so important when we switch to IMU-based heading determination. Indeed, this was also the user's recommendation. Therefore, to solve this problem, we will only use the "rotate slower" method when we are using the model to determine the reference heading.

- The participant also suggested we implement spoken instructions in the application instead of the spatialized warning-style tones currently implemented. This would entail having the application use the text-to-speech engine to provide directional words that instruct the user to rotate in a certain direction, e.g., rotate right or rotate 10 degrees. Recalling that Ross et al. [53] tested this approach and found it had the worst performance, we decided not to implement it for the third iterative experiment and leave it for future work to test.

- The next two observations point to the deficiencies of the model in its current state.

  - At the second intersection, the model was incapable of stabilizing on the correct heading as it oscillated between the three central actions (3, 4 and 5) even though the participant held his position still. As a result, the user was confused as to what he was doing wrong and kept trying to rotate to re-adjust it. An example of this is shown in Figure 4.15. The image shown in Figure 4.15a represents a scene that was relatively unchanged, i.e., little rotation by the user, and the corresponding 82 predictions from the model, which oscillated, are shown in Figure 4.15b. Note that these predicted values have already been filtered through the Kalman layer. We would expect the behaviour of CNNs to be deterministic as they consist of a series of frozen matrix multiplications. However, deep neural networks have recently been found vulnerable to well-designed input samples, called adversarial examples, where small perturbations in the input can provide substantially different outputs [99]. Adversarial defenses remains an active area of research. One way to address these issues is to draw from research in Bayesian deep learning [100], providing an additional output that indicates the uncertainty of the predictions. We leave the development of such models for future work, and, in this work, we re-train the model with additional data as well as modifications in the architecture, which we present below.



(a) Scene as presented to the model.

(b) Corresponding 82 predictions.

Fig. 4.15: An example of the model having 82 different predictions for the same input image.

- At the third intersection of the experiment, the model predicted that the user was correctly oriented. Figure 4.16 shows that input image. As we can see, the user is clearly not properly aligned with the crosswalk, which points to the deficiency of the model.

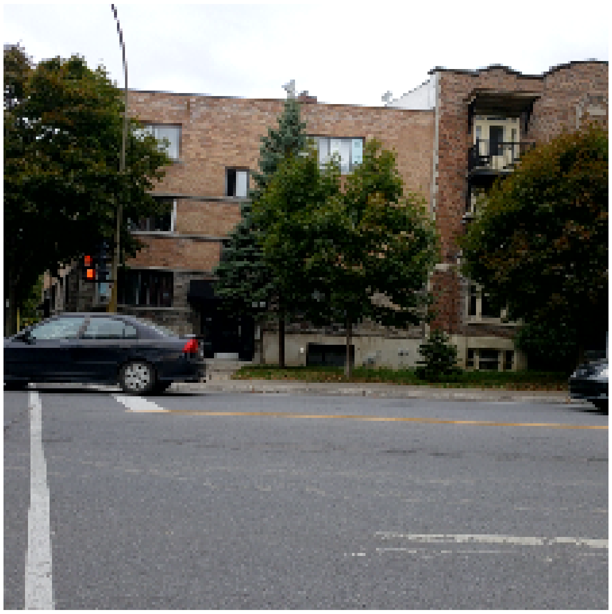


Fig. 4.16: A scene for which the model incorrectly found the heading that should be followed by the user. The correct heading would be found by rotating the user to the left as it would allow the model to locate the curb at the end of the crosswalk.

These problems prompt us to delve deeper into the model to find what causes these issues. As we explained in Section 4.2.2, using the mean absolute error was a more meaningful indicator of the model's performance. However, we originally trained all models using a classification objective. We hypothesized that re-training them to reduce the mean absolute difference would provide substantial improvements. This was accomplished by transforming the classification task into a regression task (see Section 3.1 for details of the difference). Since the *unknown* action has no spatial relationship with the other possible actions, we cannot include it as a possible action in this new objective. We address how we include the *unknown* action in this new model later in this section. Indeed, after modifying the objective and re-training the model, we tested it on the original test set from Section 4.2.2 and obtained a mean absolute error of 0.49 ± 0.05 (95% confidence interval). This is an improvement compared to the original classification model that obtained 0.71 ± 0.08, which also did not include the *unknown* action in the calculation of the mean absolute error as noted in Section 4.2.2. These values represent the distance between the model's prediction and the true (optimal) action determined by sighted experts.

After re-training the model with the new objective, we proceeded with the collection of 50 more training videos from different intersections than the original dataset. Following the strategy in Section 4.1.2, we extracted the expert's optimal crossing direction by using a single sighted individual, and we re-trained the model with the newly acquired data. The inclusion of the new data provided marginal improvements (mean absolute error of 0.47 ± 0.05 on the original test set), pointing to the fact that more data may not always result substantial improvements. To obtain a glimpse of the improvement in the model, we also tested it on the image in Figure 4.16, which was not part of the new training data. The model predicted action 1, i.e., the user should rotate left to locate the curb on the opposite side of the intersection, the correct action that should be taken.

With this new model, we have no way of informing the user that the view is blocked, i.e., no way of rendering the *unknown* action. Since the initial model appeared (empirically) to accurately predict an *unknown* situation, we use it in the mobile application as a binary classifier with its goal being to determine if the view is *unknown*. However, running these two models in series would mean that predictions would be provided with a delay of approximately 450 milliseconds, which is too long for this task. In order to balance the latency of the application, we run inference using the *unknown* classifier every three runs of inference with the orientation model. If a scene is classified as unknown, then we enter a mode where only the unknown model is used until it predicts otherwise. This schedule was chosen through empirical testing. Future work can explore multi-task learning where the two objectives would be to classify a scene as unknown, and to predict the location of the goal.

4.5.3 Third Round of Improvements

We conducted a third experiment with the same blind colleague following the improvements from Section 4.5.2. At the intersections where the model had incorrectly predicted the location of the curb or where it continuously oscillated around the correct heading, the user was now able to quickly and correctly become oriented. However, there was one minor issue we discovered during the last crossing.

At that intersection, the participant's starting orientation, i.e., before starting the application, was pointed towards the outside of the intersection, similar to the scenario depicted in Figure 4.2b. In determining the initial correct heading, the model instructed the user to keep rotating towards the outside until he was eventually facing the sidewalk. At that point, the model provided ten action-4 predictions and, therefore, transitioned to using IMU data. We believe that the model rotated the user almost 90 degrees due to the way we trained it to search for the opposite corner when it is not in the frame, i.e., by choosing the left-most or right-most spatial action continuously. This was odd behaviour, however, since the training and testing data included examples where the user was oriented towards the outside of the intersection, as is the image in Figure 4.2b. This points to the need of further improvements to the model architecture, and/or further efforts to extend the datasets to include more examples of this configuration. As for the model predicting the correct heading when it faced the sidewalk, this behaviour can be explained due to the training data. When we extract frames from the initial crossing videos to create the dataset, the last few frames always included views of the sidewalk since we are at the end of the crossing. And these frames were labelled with the central bin since it is the correct heading the user should follow. These effects can be mitigated by excluding images from the end of crossings from the dataset. This intuitively makes sense since we are only using the model for initial orientation. We note that although this behaviour is undesirable, it is does not pose a safety issue. We discuss further improvements to the model in Section 4.6.

Interestingly, the user did not realize he had rotated almost 90 degrees to the outside. We note that this would probably not have occurred if the user was not using the application, since he would have relied on his orientation and mobility training instead. We had not observed users putting more trust in the application during our experiment with blind participants (presented in Section 4.4.4), which is understandable since the experiment's time span was at most 1.5 hours. However, since this participant had now become a frequent tester of the application (having tested it on three separate occasions), it would seem he has now reduced his reliance on his own abilities, and has put more trust in the application. Although anecdotal, this shows the importance of providing accurate feedback to users in the assistive technology domain as their reliance on these applications can increase with time.

4.6 Limitations and Future Work

While the application we designed produced good results overall, we note that there still exist areas for continued development. One area of improvement lies in the design of the final layer and the loss function. While the new model we proposed in Section 4.5.2 yielded improvements in the overall mean absolute error of the predictions, we hypothesize that this can be further improved by applying methods from ordinal classification [101]. This new objective offers a way of performing standard classification while taking into account the order of classes. That is, a higher loss is obtained if the model predicts a class (e.g., action 6) farther from the true class (e.g., action 3) compared to if it predicts a class (e.g., action 4) closer to the true class. This is precisely the behaviour we desire from the model if we remove the *unknown* class, which has no spatial relationship to the other classes. Using methods from multi-task learning [102], one can design an architecture with one task being to determine if a scene should be classified as *unknown*, and a second task that predicts the optimal direction, given that the scene is not *unknown*.

Another area of improvement is the design of the architecture. A decision we made was to use single-state-single-action imitation learning, i.e., taking into account a single timestep to make a prediction. That is, in making a decision, the model only looks at the most recent image. Future work should experiment with using the sequence of prior frames of the current crossing combined with the current frame to decide on the optimal action. This can be accomplished by employing a recurrent neural network, where at every timestep, it obtains features of the current image through a CNN, and the hidden state vector holding information from all previous inputs. Further design efforts would be needed to make such a model operate efficiently in a mobile application.

Analyzing the training/testing losses can result in better model performance. Even with the improvements we applied and the additional data we collected (Section 4.5.2), the model still seems to overfit to the training data. This could be due to the model having too much capacity allowing it to learn visual features specific to the training data [103]. We can improve the model's performance by training it with more data. After the release of this work and the improvements we proposed in Section 4.5, we discovered the Freiburg Street Crossing Dataset [104], which contains images from first-person perspective originating from the corner of intersections, similar to the ones we collected. Future work should employ sighted individuals to label the images from this dataset with heading targets following the instructions from Section 4.1.2. We expect this additional training data will reduce the overfitting we are observing.

It is important to note that we constructed this dataset by learning how visually impaired individuals cross intersections through observation sessions with only two users. While this has provided us with an overall understanding of the problems, it does not represent the full range of strategies employed by visually impaired individuals in intersection crossing. The procedure employed in this chapter to construct a dataset may need to be repeated prefaced by more observation sessions with individuals from the blind community. These sessions can be further enhanced by working closely with orientation and mobility specialists who train blind individuals in outdoor navigation.

In designing computer vision applications for people with visual impairments, an important consideration is the image acquisition method. The body harness used in Section 4.5 (shown in Figure 4.13) provided a stable method of holding the smartphone for camera-based computations. While the participant of Section 4.5 preferred this harness over the original lanyard design, shown in Figure 4.6, we hypothesize that this may not be the case with other blind individuals due to cosmetic acceptability and social impact. Further research should explore user preference between these two approaches as well as using glasses with an embedded camera, such as the *Horizon Smart Glasses* developed by Aira Tech Corp [105].

Following the improvements of our application from Section 4.5 and the modifications to the model discussed above, more experiments with blind individuals should be conducted. These experiments can also explore different camera positions to answer the questions of user preference, and to find out how the position affects the performance of users. As opposed to the experiment we conducted in Section 4.4, these experiments should be conducted at more than one intersection to test the system's robustness. In addition, since the amount of traffic is an important signal that visually impaired individuals rely on for orientation and crossing, future studies should record the number of cars during trials; this is one of the limitations of our study. Another consideration for subsequent studies is to conduct them with only one type of assistive device, i.e., guide dog or white cane. As we have seen with one participant who used a guide dog, the application may provide feedback that opposes the guide dog's guidance.

During our experiments of Section 4.5, it was important that we balanced the tension between customizing the system to one user's preferences and ensuring that the system is generalizable. These preferences were expressed by the volunteer when he requested spoken instructions rather than audible tones, or when he indicated that the body harness is a better alternative to the original lanyard. In the current stage of the application, it is important that the basic components for functionality (i.e., the model) perform well in the simplest testing scenarios. These preferences can then be addressed with a more comprehensive pool of participants to ensure that the system can meet the preferences of the community as a whole.

The objective of this work is to eventually release a mobile application on smartphone application stores. For this application to be useful for the visually impaired community, it will need to tackle all the challenges involved in crossing an intersection, as identified in Chapter 1. Radwan et al. [104] propose a multimodal CNN architecture for predicting an intersection's safety by jointly predicting the state of the traffic light and the future trajectories of surrounding traffic participants. Future work can explore the integration of their work with our approach, and intersection configuration data (e.g., two-way vs. four-way) obtained from *Open Street Map*, to provide a solution that tackles all subchallenges of intersection crossing.

Chapter 5

Describing Visual Content to Blind Individuals

The recent surge of deep learning has presented new opportunities in understanding and describing visual content, with models capable of classifying objects in images [17, 16, 18, 1] and providing natural language descriptions of those images [42, 43, 2, 22]. The inception of these models has led many to believe that they are ready for deployment in applications tailored for the blind community. Although the potential they offer is attractive, further research is required before such captioning models are ready to be applied for the visually impaired community. We attribute the problems with these models in part due to the disparity between the data used to train these models and the requirements of visually impaired users. In addition, the frequent inaccuracies of current models make them unreliable and even unusable by this user group. In this chapter, we review previous approaches that study describing visual content to visually impaired people. We then summarize the lessons learned by these various methods, with emphasis on the type and quality of the descriptions required as well as the human-computer interaction challenges associated with obtaining images from the blind user's perspective. Finally, we suggest future directions we believe researchers and AI practitioners should follow when designing scene description systems for this community.

5.1 Assessment of Current Models

We start with the presentation of results from recently proposed image captioning models, shown in Table 5.1. *CaptionBot* [5] is an open API powered by Microsoft Cognitive Services, and provides image captions that start with "I think" to promote skepticism in the captions. *Neural Image Captioner (NIC)* is the model proposed by Vinyals et al. [2], which we described in Section 3.1.3. We can see that only the first example in Table 5.1 provides an accurate and satisfactory description of the image. From our experience, the errors we encountered here occurred frequently throughout our testing of deep learning models on images which were acquired outside of the training distribution, i.e., images outside of the Microsoft COCO dataset [44].

Firstly, although the models are partially correct in most of the cases, as we can observe, the amount of detail provided is limited. Furthermore, the additional detail they provide is typically the source of the errors. This can be observed, for example, in the second image-captions pair where both models make the error of predicting the presence of traffic. As we have previously discussed, these models often provide repetitive captions. This can also be observed by comparing *NIC*'s captions in the second and third examples. Finally, some of the errors make these models unusable by the blind community as users would likely be led to potentially dangerous situations. As we mentioned in Chapter 1, one of our *Autour* users indicated that a potential use case for these models would be to identify sidewalk closures and other roadwork. We see in the last example of Table 5.1 that users would be misinformed by the generated captions.

The incorrect "assuming" of the presence of an entity, while understandable in most cases, suggests that the models associate specific entities in the image (street) with other details not necessarily present (cars). This was a recurring source of error throughout our experience, and can be observed in almost all the image-captions examples. We hypothesize that these errors are due to training the models with a maximum likelihood objective, which, by definition, provides for the most likely prediction. However, without properly grounding the caption generation process in the image through an alternative objective function, it seems inevitable that the models will make such errors.

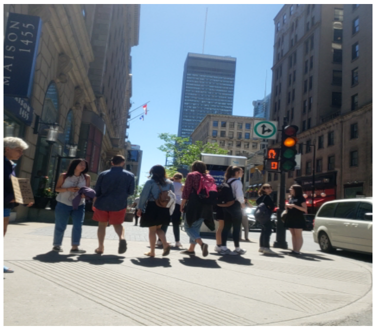
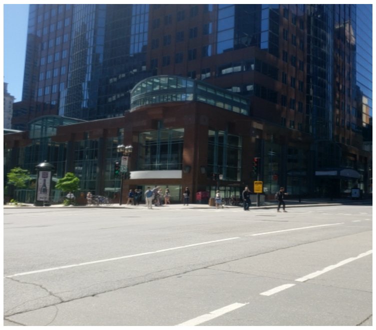
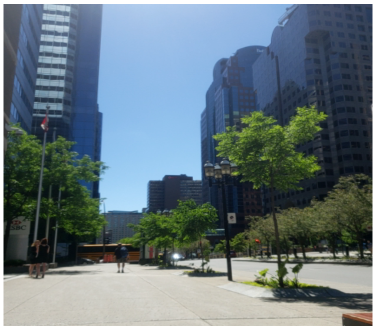
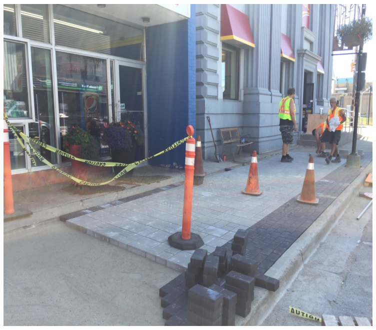
Image	Descriptions
	• **CaptionBot:** I think its a group of people walking on a city street. • **NIC:** A group of people walking down a city street.
	• **CaptionBot:** I think it's a close up of a busy city street. • **NIC:** A city street filled with lots of traffic.
	• **CaptionBot:** I think it's a view of a city street. • **NIC:** A city street filled with lots of traffic.
	• **CaptionBot:** I think it's a fire hydrant on a city street. • **NIC:** A red fire hydrant sitting on the side of a road.

Table 5.1: Images taken while walking in Downtown Montreal with a smartphone worn in a lanyard as in Figure 4.6. **CaptionBot** captions are generated using the CaptionBot API [5]. Neural Image Captioner (**NIC**) is a trained version of the model we presented in Section 3.1.3, which is based on the model proposed by Vinyals et al.[2].

5.2 Survey of Previous Work

In recent years, many systems and research projects have been proposed to assist visually impaired people with the understanding of visual content, whether in indoor and outdoor environments [39, 29, 106, 7, 25, 24, 107, 26] or on social media platforms [108, 8, 9]. In this section, we describe the projects that performed a formal analysis of their proposed systems with blind individuals, describing how they analyzed the problems, the solutions they propose and the lessons they learned from their studies.

Bigham et al. [29] designed *VizWiz*, a crowdsourced iOS mobile application that allows blind users to obtain answers to questions regarding images, which are taken from their phone, by asking multiple volunteers on the web. The authors also introduced *quikTurkit*, an approach to intelligently recruit sighted workers in advance with the goal of reducing latency. When a user starts the application, *VizWiz* pings the server to start recruiting participants in anticipation of an incoming question. *quikTurkit* maintains a pool of workers by displaying questions that were asked previously to keep them busy until the new questions arrive. *quickTurkit* accomplished this seamlessly by posting previous Human Intelligence Tasks (HITs) on the Mechanical Turk server. The workers interacted with the image and question through a web page. The authors conducted multiple user studies with blind individuals to assess and iteratively improve their application. An interesting finding was participants' frustrations with being unable to take good pictures, with 11% of questions being discarded due to image problems, which was later partially addressed by the group in their second study by means of detecting blur and darkness. After analyzing the questions typically asked by users, the authors found that many questions were motivated by the users' desire to locate a certain object. The authors proposed *VizWiz::LocateIt*, a prototype system that combined the *VizWiz* question-answering approach with computer vision to help users locate certain objects in an image. The system follows an information visualization scheme that consists of an overview stage, followed by a zoom and filter stage, and allows for details on-demand. To assess their method, the authors designed a within-subject lab-based study where participants were asked to find a desired cereal box using *LocateIt* and a commercially available barcode scanner with a talking interface. While their study showed that *LocateIt* was slower and less accurate than using the barcode scanner, its advantage is that it has the capability of scaling to objects that do not have a barcode. Their study revealed that some participants had difficulties keeping the phone perpendicular to the ground, and that all participants had difficulties judging how far back they should hold the phone from a cereal box when they tried to center it in the frame.

After a one-year deployment of *VizWiz Social*, an application derived from *VizWiz*, Brady et al. [30] analyzed 1,000 questions asked by visual impaired users, which were sampled from a dataset of 40,000 questions posed by 5,000 users. While *VizWiz* can only send questions to Mechanical Turk, *VizWiz Social* users can employ different answer sources such as VisionIQ, a computer vision service that uses human workers to manually identify unrecognized objects, or use friendsourced workers using email or Facebook. Their analysis provided insights into the common question types typically asked, and helped understand the photography challenges faced by this user group. We present the four main categories the authors found in Table 5.2.

Category	**% of Questions**	**Description**	**Subcategories**
Identification	41%	An object to be identified by name or type.	No Context, Contextual, Medicine, Currency, Media
Reading	17%	A user requests the text be transcribed.	Information, Mail, Digital Displays, Number, Bathroom, Cooking
Description	24%	A user requests a description of visual or physical properties of a depicted object.	Appearance, Color, Clothing Color, Clothing Design, State(on/off), Computer/TV Screen
Other	17%	A question that was unanswerable.	Outside of Range, About VizWiz, Unanswerable (e.g., audio issues)

Table 5.2: Categories of questions asked by *VizWiz Social* users after a one-year deployment. Percentages do not add up to 100%.

A rater placed each image-question pair into one of the categories in Table 5.2, and then further into one of its sub-categories. We can see that most of the questions were related to the identification of an object, e.g., identifying what kind of *soda can* the user is holding or what object is shown in the picture. The second most common question type was the description of certain scenes such as how old a person looked or the color of the flowers shown. The authors also performed a classification of the images alone without having context from the question, which provided insight into the categories users were most interested in accessing. They found that 76% of the photographs fell into the *Object* category while 5% and 4% focused on *Person/Animal* and *Setting* categories, respectively. Next, an evaluation of the photograph quality was conducted based on blur, lighting, framing (are parts of the item in question outside the frame?), and composition (is an item obscured by other objects?). Using a researcher, each photograph was initially given a score of 5 with a point being deducted for each error found until the minimum score of 1. They found that only 15% of photographs obtained a perfect score, with 33% scoring 4, 29% scoring 3, and an average score of 3.41. Despite the errors however, only 5% of photographs with scores of 3 or 4 were determined not to have an identifiable object. It would be interesting to see if this recognition rate would differ with deep learning approaches. In addition to analyzing the questions and images, the authors also examine the behaviour of users to offer insight into the challenges and successes of adopting access technology for blind individuals. Analysis of the 25 most active users was conducted, who asked an average of 283 questions through an average of 295 days of usage. For each user, the authors analyzed the first and most recent five questions. The first five questions' analysis showed that the majority of questions related to *identification* (73%), followed by *Description* (14%), *Other* (9%), and *Reading* (4%). The latest five questions consisted of 46% *Reading*, 25% *Identification*, 21% *Other*, and 8% *Description*. In addition, they found a significant improvement in photograph quality from an average of 3.32 to 3.62. The authors also found a higher-than-usual abandonment rate when users had a poor first experience either due to poor quality of their question-image inquiry or a poor answer from the crowd, pointing to the importance of usability as well as utility in the adoption of assistive technologies. They provided more insight into the urgency of questions as well as the level of subjectivity vs. objectivity of the questions for which we choose not to report in this thesis. In 2018, the *VizWiz* application resulted in the release of a new visual question answering dataset [45] tailored for blind individuals, sparking a new computer vision challenge held at ECCV 2018 [46].

One of the limitations of *VizWiz* is that the user can only send images to a group of workers one at a time, which makes it ineffective in supporting users over the course of a sequential interaction. Lasecki et al. [106] addressed this with *Chorus:View*, a system designed to enable the user to have a consistent and reliable conversation with the crowd through a video stream from the user's phone. To motivate the use of video streaming, they analyzed a one-month period of questions asked on *VizWiz Social* and found that 18% of those questions are likely to require a sequential interaction. The sequential requirement related to a scenario where a user needs to ask multiple questions about an image that cannot be answered in one shot, or a scenario where photographic errors require corrective instructions from a crowdworker before an answer can be provided. They conducted multiple user studies that tested blind users' ability to use the *Chorus:View* approach in completing a set of tasks in comparison to *VizWiz*. In a preliminary study that used a Wizard-of-Oz design, six blind users were paired with six students (workers) where users submitted questions by streaming video using the Apple FaceTime application, and workers replied to questions via text messages that are then read by VoiceOver. The results showed that users spent less time finding the information using the *Chorus:View* approach, and on a 7-point Likert scale where 1 was "strongly disagree" and 7 was "strongly agree", users rated the approach of *Chorus:View* to be easier to use than *VizWiz* with a Likert score of 6.0 versus 3.0.

Next, we review some details of the *Chorus:View* application. Users stream video through the application and record their questions via audio at any time throughout the stream, and the workers' feedback to those questions is provided in a text area that is automatically read using VoiceOver. As users had difficulties stabilizing the camera, the interface allowed workers to capture individual frames from the video stream, helping them focus on specific details. In addition, the worker interface allowed multiple workers to contribute to an answer by allowing them to interactively vote on existing responses as well as propose new ones. Workers were not only rewarded for quickly generating answers but also for coming to agreement with others. A second study was conducted with 34 distinct crowd workers and found a significant improvement in the speed and accuracy of answers generated by five workers compared to a single worker. The authors conducted a third user study with ten blind participants and 78 unique crowdworkers where the tasks were the following:

- *Product Detail*, accomplished by finding a specific piece of information about a product, such as expiration date.

- *Sequential Information finding*, accomplished by asking users to find a package of food which is not identifiable by shape.

- *Navigation*, by simulating the finding of an accidentally dropped shirt.

The authors found a significant improvement in response time and accuracy when compared to *VizWiz*. Moreover, with a set limit of 10 minutes for each question sequence, *Chorus:View* had a significantly higher completion rate (95%) than *VizWiz* (40%). Users expressed enthusiasm and excitement regarding the potential usefulness of *Chorus:View* while stating that *VizWiz* is also useful for non-sequential tasks. Interestingly, the authors also reported feedback from workers, noting that although it is rare for Mechanical Turk workers to provide positive feedback after their study, they received multiple emails expressing workers' desire to continue participating as they felt good contributing to helping others.

As *Chorus:View* uses a video stream, it can potentially be very expensive using up the data quota, as noted in [106]. In addition, a video-based approach can be difficult for the user who must wait, holding the camera steady, while the crowd determines a response [7]. The goal of *RegionSpeak*, proposed by Zhong et al. [7], was to account for many of the tasks that fall between what can be solved by single-image approaches, e.g., *VizWiz*, and the continuously-engaged interaction, e.g., *Chorus:View*. Their approach allowed users to capture more information per interaction by stitching multiple images together, as well as having the crowd workers focus on describing specific regions in the stitched scene using natural language sentences. Users explored the scene by moving the camera in any direction and, using a key frame extraction algorithm, the application instructed the user to hold their position while it captured a new image automatically. The application sends three to six frames, which was determined through a pilot test, to a remote server that performs the stitching. The authors conducted two main experiments to test the usability and effectiveness of different parts of their applications. The first confirmed that their stitching interface was easy to learn and use by blind people, with an observation that users felt it easier and less stressful to follow the application's framing instructions versus following crowd workers' instructions, which caused them to retake photos several times. Their second experiment, which consisted of three parts, dealt with the elicitation of visual descriptions. The first part introduced five experimental images extracted from 1000 questions from *VizWiz* that could not be answered, and used in all future experiments to evaluate *RegionSpeak*. This part employed ten Mechanical Turk workers who were asked to describe the content of the image with a single sentence, providing a total of 50 datapoints. The authors evaluated the quality of those descriptions using the following metrics:

- Validity: if an object is described as being in the image, is it actually shown?
- Minimalist: Does the answer appear to be the answer requiring the least effort?
- Distinct Items: How many distinct items are named in the answer?
- Details: How many explanatory details are provided beyond the core description?
- Spatial Information: How many spatial cues are provided in the answer?

In the second part, the authors selected the most complex scene (an outdoor scene) and explored increasing the descriptive level of the answers through an iterative process. This was accomplished by iterating over the ten initial descriptions provided by the first part of the experiment three times with crowd workers. After the iterative process, the authors found a significant improvement in the number of objects described, details provided and spatial information provided, and found a near-significant decrease in the number of minimalist answers compared to the answers from the first part of the experiment. One problem with the iterative process was the time costs of generating the final description, which can take four times longer to return to the user than with a single description author [7]. Another problem was the fact that crowd workers' workload was not equal and difficult to predict since each workload depends on the quality of the previous worker's description. The third part of the experiment addressed this issue where the authors propose a parallelized approach to providing labels for different objects in the image. The crowd workers were asked to select the object through a rectangular selection tool, and provide a description of it. All crowd workers were able to see a list of the set of objects that others had labeled up to that point. The final description that a user received was a stitching of all the descriptions received, each associated with a specific area. Unfortunately, the authors did not provide a direct comparison with the iterative process from the second part, which had only used one of the initial five images. The authors note that while workers were able to see the set of objects that others had finished labelling, this did not guarantee the prevention of duplication of labels being provided simultaneously unless each label was obtained in series, which would result in the same latency as the iterative approach. Ideally, *RegionSpeak* would maintain a shared state that shows the workers' annotations in real-time.

Wu et al. [108] proposed the automatic alt-text (*AAT*), a method that provides blind users on Facebook with automatically generated captions describing images. The advantage of using machine-generated captions is that they can be provided in real-time, which improves the browsing experience of people with visual impairments. In their work, the authors trained a model on 97 concepts that are made up of objects and themes. These concepts were selected based on their prominence in Facebook photos, which was determined by having 30 human annotators label three to ten things in each photo. After selecting the 200 most frequent concept candidates from these photos, and through further filtering, the authors selected the 97 concepts that covered the major categories of tags. The authors performed an in-lab study and a field study. Interestingly, the in-lab study showed that blind users preferred to hear the constructed alt-text in a form of a complete sentence as it is more natural and friendlier. However, due to limited accuracy and consistency of current image captioning systems, the authors decided against a free-form sentence model and instead opted for a fixed sentence that started with "Image may contain:", followed by the list of tags. As the authors noted, "may" was used to convey uncertainty. The in-lab study showed that all participants would like to have more tags, with one user explicitly indicating that she'd rather have more detail even if it means the accuracy of the information might be incorrect as opposed to having no information at all. Noting that balancing between the amount of information provided and the risk of wrong information is a limitation of *AAT*, the authors applied a threshold on the tags choosing to render concepts that obtained a confidence score of 0.8 or higher for a smooth experience. This work also included a two-week field study with 9000 visually impaired Facebook users separated into a test group, for which images were annotated with *AAT*, and a control group who used Facebook as they normally did. This study showed that the test group found Facebook more useful to them, that photos were easier to interpret, and users indicated they were more likely to socially interact with the photos. However, the last of these was not found when the authors analyzed the logged data of the users. An analysis of the write-in feedback was conducted, classifying a given feedback into three themes: *useful*, *not useful*, and *improvements*. While over 90% of the feedback was classified as useful, the authors note that it is a biased feedback since this user community might provide positive feedback to show appreciation for accessibility efforts. Some users had found *AAT* not useful, pointing to vagueness or accuracy issues of the descriptions. In general, users who pointed to improvements focused mainly on two avenues: extracting and recognizing text, and obtaining more details about people including identity, age, clothing, action and emotional state. Finally, the authors noted that future work should not only identify objects and themes but also say something about their relationship, similar to natural descriptions provided by *RegionSpeak*.

Macleod et al [8] noted that most image captioning systems have been evaluated based on how well they correlate with sighted individuals. Instead, the authors focused on assessing blind people's experiences with automatically generated image captions, specifically focusing on the role of phrasing on a blind person's trust in a caption. They conducted an initial user study with six blind participants to understand their current experiences navigating images on Twitter and their feeling of computer-generated image captions. Participants were directed to an artificial account made by the authors with 14 images, an associated tweet text and a computer-generated caption obtained from *CaptionBot* [5], which provides some level of uncertainty in its descriptions. Of these samples, six captions were accurate and complete, four were accurate but lacked information, and four were completely wrong. The authors found that participants: (1) would assess the accuracy of a caption based on how well it matched the associated tweet text, (2) showed signs of being aware of the confidence warnings provided by the model, and (3) had a varying willingness to encounter wrong captions.

To further understand blind individuals' experiences, the authors conducted an online experiment with 100 visually impaired participants. Participants were again presented ten tweets that contained an image, a tweet text and a generated caption. After the presentation of each tweet, to assess their **understanding**, participants were asked to rate on Likert scales the extent to which the caption (1) improved their understanding of the image, (2) improved their understanding of the tweet as a whole, and (3) would be helpful to other visually impaired people (a method of eliciting truthful subjective data [109]). After experiencing all ten tweets, to assess their **overall trust**, the participants were questioned regarding their trust in the caption, and asked to rate the intelligence of the computer algorithm generating the descriptions. Finally, to assess the amount of **detail**, participants were also asked if they would like more information on any images, and **why** or **why not**. As stated above, the authors wanted to learn more about framing effects, i.e., how does the phrasing affect trust and understanding. They varied the framing by providing the confidence either as numerical or in natural language, and by providing it positively or negatively ("10% chance that's..." *versus* "90% chance that I'm wrong but

I think...”), resulting in four possible study groups to which participants were randomly assigned. In addition, they selected images that allowed for a range of **congruence** with the tweet, i.e., how well the caption matched the user’s expectation based on the tweet text. Some of the results presented by the authors are the following:

- Participants, as a whole, found the captioning system to be fairly trustworthy and useful in understanding the tweet regardless of the study group, with scores slightly over three on the five-point Likert scales.

- Participants found tweets with high congruence to be more useful and more trustworthy, as one would expect.

- Negatively framed captions with **high** congruence were more helpful to understanding the image then negatively framed ones with **low** congruence.

- Positively framed captions with **low** congruence were more helpful than ones with **high** congruence, suggesting that positively framed captions are perceived as adding to the understanding of the image when there is a mismatch with the tweet text.

- Analyzing low congruence captions only, participants receiving negatively framed captions trusted them significantly less often than people who received positively framed captions.

- On framing, the authors found that participants trusted tweets with high reported confidence significantly more than tweets with low reported confidence. Although this result is intuitive, it is important to note that the reported confidence, which is obtained from the model, does not always align with the accuracy a human evaluator would assign.

- Participants found tweets with reported low confidence and negatively framed captions significantly more detailed than ones with positively framed captions.

Salisbury et al. [9] investigated ways of combining crowd input with existing automated image captioning approaches to assist blind people in accessing visual content on social media. The authors designed four workflows for providing understanding of images with associated tweets to visually impaired users. The first workflow used *CaptionBot* [5] alone to provide a caption of the tweet’s image. The second workflow, called Human-Corrected

Captions, provided crowd workers with the tweet text, the image and the caption obtained from *CaptionBot*. The worker must then improve the caption given the context of the tweet text and image with the goal of explaining the image to a blind user. The third workflow, called the TweetTalk conversational assistant, built on the two other workflows by providing blind people, or simulated blind people, a conversational assistant platform. This workflow connects two workers together, one who plays the simulated visually impaired user (SVIU) and the other plays the sighted assistant role. These workers then followed these steps:

1. **Read the tweet**: Both workers are shown the tweet's text and baseline image caption, which can be empty or can originate from the first or second workflows. The "sighted assistant" is the only one shown the image associated with the tweet.

2. **Rate the caption**: The SVIU is asked to rate the baseline caption, if there is one. This provides an initial assessment of the blind user's trust and usefulness.

3. **Ask/Answer questions**: With access to a chat box, both workers participate in a question-answer conversation regarding the image.

4. **Write a Description**: Following the rounds of questions, the SVIU then generates a new description of the image, providing an understanding of their gained insight.

5. **Feedback**: The SVIU is shown the image and asked to rerate the baseline caption and the new one she generated in step 4.

In rating the captions (steps 2 and 5), the SVIU was asked the same question as Macleod et al. [8], i.e., "I think visually impaired people would find this caption helpful" on a Likert scale. The time it took two workers (one SVIU and one assistant worker) varied in this workflow ranging from 2 to 20 minutes with an average of 8 minutes.

The fourth workflow, named Structured Questions workflow, was streamlined based on the most common question types asked in TweetTalk Conversation Assistant. This workflow followed the same procedure as TweetTalk Conversations replacing step 3 with structured questions, presented in the following list:

- Who are the main subjects of the image (people, animals, notable objects, etc.)? Describe their physical characteristics (notable features, clothes, poses, relative positions, etc.)

- Where is this set? Describe the location and the prominent features of the background.
- What are the subjects of the image doing? Describe their actions, and their intent.
- What emotion does this image evoke? Or what are the emotions of those present in the image?
- Describe any noteworthy aspects of the images visual style.
- Is this tweet intended to be humorous? Explain how.
- Is this a famous or well-known image?
- Does this tweet contain a meme (meme images, hashtags, etc.)? If so, describe what the meme is about.

The first four items from this list are relevant to describing general environments, i.e., environments not particular to social media platforms. Given this list of questions, the "sighted" workers decided on whether or not each question was useful given the tweet text and image, and provided an answer to the questions they found useful. The question with the most votes was chosen as the source question, and then the longest answer for that question was provided to the user (The authors note that other mechanisms are possible). Note that the Structured Questions workflow was easier and faster for recruiting workers as it did not require the pairing of workers. Through the structured questions, the time it took for workers to answer questions ranged from 3 seconds to 14 minutes with an average of one minute.

To assess the baseline captions and the descriptions generated through the workflows, the authors conducted an experiment using crowdsourced workers. We want to emphasize the type of data obtained from their experiments. If a baseline caption was provided from the Vision-to-Language model or the Human-Corrected model, that caption received two Likert scores from the SVIU, one at the very beginning of the workflow (whether TweetTalk Conversations or Structure Questions) *before* seeing the image, and another score at the end of the workflow *after* seeing the image. The descriptions generated from step 4 of the third and fourth workflows were also rated by the SVIU *after* seeing the image. In addition, a third-party worker, who had no knowledge of the conversation, was employed to rate the captions *before* and *after* the workflows. The authors refer to ratings provided by the SVIU who participated in the conversation as first-party ratings, and ratings provided by the third-party worker as third-party ratings. We present some of the results found in their study:

- Regardless of the seed caption (or no caption), the descriptions generated by users working through the TweetTalk workflow, with or without the structured questions, had no significant difference in rating when assessed after the users viewed the image.
- In the generated descriptions workflows, first-party ratings were consistently higher than third-party ratings. After confirming that the disparity was not due to first-party raters having gained intrinsic value through the conversation, the authors suggested that the disparity is due to workers rating their own generated descriptions higher than they should.
- AI-generated captions had significantly worse accuracy then any other caption source, i.e., human-in-the-loop corrections, Conversational Assistant or Structured Questions.
- Structured Questions workflow improved understanding of the image better than all other approaches, according to the third-party ratings.
- Seeding conversations with AI-generated captions resulted in significantly less satisfaction by first-party and third-party raters *after* having seen the image. The authors note that these findings are due to blind users (or SVIU) initially placing too much trust in AI-generated captions, as found by Macleod et al. [8], and that the inaccuracies of these seed captions often led people astray in properly framing questions.
- Time cost of answering structured questions is significantly lower than going through step 3 of the TweetTalk Conversational Assistant workflow.

Although their study employed simulated blind workers, the authors validated the list of questions through a subsequent study with seven blind participants using TweetTalk (the third workflow), finding no significant difference in the questions asked. In addition, using third-party sighted raters, there was no significant difference in the ratings of the generated captions of step 4 in the TweetTalk Conversational Assistant workflow between the blind participants and the simulated blind workers from the first study.

5.3 Lessons Learned

The literature we reviewed provided us with a glimpse of the current state of research in describing visual content to visually impaired people. In this section, we provide a summary of the lessons we learned through this review. We begin by focusing on specific requirements for providing general image descriptions to blind individuals, with the assumption that the knowledge from research in social media descriptions can be transferred to general descriptions. Next, we provide a summary of successful ideas employed in crowdsourcing for visually impaired individuals. Finally, we analyze the general challenges faced by this community in taking good, stable and properly framed images with the goal of obtaining more information about them.

5.3.1 Description Requirements

Through the review of the literature in Section 5.2, we can identify the most important features that an image description should contain for a blind user. As previous research suggests [29, 30], the identification and localization of objects is an important feature for this community. In rendering the objects to users, one could take the approach followed by Wu et al. [108] by stating a start message "Image may contain:" followed by a list of objects contained in an image. This kind of visual description was useful for blind Facebook users, resulting in an increase of their likelihood to participate in social media picture sharing, commenting and liking. However, as they note, if automatic image captioning systems can provide accurate and consistent image descriptions in fully-formed natural language sentences, the experiences of blind individuals would be more natural and friendlier. Visually impaired people desire descriptions that not only provide a list of objects and visual concepts, but also provide information about their relationships. Contextual information such as identity, age, clothing, actions and emotional state are important parts of understanding a scene [108]. All of the questions asked in the Structured Questions workflow by Salisbury et al. [9] address many of these requirements, with half of them being relevant to describing scenes not restricted to social media (e.g., Tweets). We provide these relevant questions in Table 5.3.

Question
Who are the main subjects of the image (people, animals, notable objects, etc.)? Describe their physical characteristics (notable features, clothes, poses, relative positions, age, etc.)
Where is this set? Describe the location and the prominent features of the background (weather conditions, architecture of surroundings, landmarks, etc.).
What are the subjects of the image doing? Describe their actions, and their intent.
What are the emotions of those present in the image?

Table 5.3: Guideline questions to ask when describing an image to a visually impaired individual.

These questions provide substantially more details compared to the instructions subjects were provided when generating captions for the Microsoft COCO dataset. The prescriptive nature of the latter instructions, which are presented below [110], is less useful as guidelines in determining relevance for the visually impaired community.

- Describe all the important parts of the scene.
- Do not start the sentences with There is.
- Do not describe unimportant details.
- Do not describe things that might have happened in the future or past.
- Do not describe what a person might say.
- Do not give people proper names.
- The sentences should contain at least 8 words.

We note that the amount of information required may vary depending on the knowledge and desire a user might have to obtain certain types of detail, such as color [9]. The literature we reviewed suggests that the majority of blind users would prefer the maximum amount of information, even at the cost of possibly obtaining incorrect information [108]. These users then have the choice of trusting the provided information [108], and their thresholds for accepting wrong captions varies [8]. As determined by previous research projects [8, 108] and our general experience with these models outlined in Section 5.1, current image captioning systems often provide inaccurate or inadequate descriptions for visually impaired people. Even if an automatically generated seed caption is provided to a sighted conversational assistant, the blind user is often incapable of gaining an accurate understanding of the image as the caption completely misleads the conversation [9]. As Macleod et al. [8] explored and Salisbury et al. [9] found, users tend to place too much trust in automatic captioning systems. As noted above, Wu et al. [108] phrased their automatically generated descriptions with "may contain" with the goal of giving the user a sense of uncertainty in the information. The choice of trusting a caption can be made easier if the descriptions are rendered with a confidence score, which we know blind people pay attention to [8]. Ideally, scores should reflect the accuracy of the model, which is not the case with current models [8]. In determining the level of trust they should place in image descriptions, blind people tend to look for congruence with external signals, e.g., tweet text [8]. We hypothesize that they would also look for congruence between a scene description and the environmental sounds. In addition, the framing of the confidence scores has an effect on the understanding and trust placed on these systems. Negatively framing captions in situations with high congruence seems to produce the ideal behaviour by having blind individuals place the least amount of trust in the captions [8]. We believe promoting skepticism is a desirable property from automatic caption generators since they are still prone to errors and current confidence measures provided by models do not reflect their accuracy.

Moreover, from our review in Section 5.2 and Section 2.2.2, we can identify the best practices for evaluating a description's accuracy and relevancy from both a blind user's perspective and a sighted individual's perspective. A sighted individual can rate an image description based on its validity and the amount of detail it contains [6, 7]. The set of questions that a sighted rater should be asked are presented in Table 5.4.

Subject	**Questions**
Validity	If subjects are mentioned in the description, are they actually present? Are described actions of people actually occurring? Are the described relationships between objects actually present?
Detail	Are the most salient subjects (objects, persons, etc.) identified in the description? Does the description identify the relationship between all important subjects? Does the description contain information regarding weather conditions, notable architecture/landmarks, emotional state of persons, color of objects, clothes, poses, age, etc.?

Table 5.4: Guideline questions to ask a sighted individual when assessing the quality of a caption. These questions are based on previous findings [6, 7].

A blind user can rate a description based on the understanding it provides, the amount of trust they put in it, and relevancy of the details [8]. The set of questions that a visually impaired rater should be asked are presented in Table 5.5. These questions presume that the visually impaired user would provide their feedback immediately following the rendering of the description, and would, thus, still perceive the auditory scene and other contextual cues.

Subject	Questions
Understanding	Does the description improve the understanding of the surrounding? Would this description be useful to other blind individuals in understanding their surroundings?
Trust	Quantify the amount of trust placed in the description. Does this description align with what you think is going on around you based on the sound of the surrounding and why? Do you think the description provided is machine-generated or generated by a sighted individual? How intelligent would you rate the entity that provided this description?
Detail	Are the details provided relevant in understanding the surrounding? Would you prefer to obtain more or less information than what was provided and why?

Table 5.5: Guideline questions to ask a blind individual when assessing the quality of a caption. These questions are based on previous findings [8, 9].

5.3.2 CrowdSourcing Success

Ideally, image description can be provided automatically through an AI system [108]. As we discussed above, this is not a viable solution as models remain inaccurate and inadequate for the blind community [29, 7, 8]. Many successful approaches [29, 106, 7, 9] have explored the use of crowdsourcing in assisting blind people with understanding visual content. As noted by Lasecki et al. [106], sighted people enjoy being part of such projects as they feel they are positively contributing to helping others. This is also indicated by the number of blind users (102, 133 individuals) compared to the number of sighted volunteers (1, 767, 370 individuals) on the *Be My Eyes* mobile application.

All the crowdsourcing approaches we explored in Section 5.2 employed multiple workers in providing an answer. This allows the user to obtain different workers' perspective on an answer [29] or improves the quality of the answer [106, 7]. Bigham et al. [29] provide responses to user questions by sending the request to multiple workers who must each provide an answer. In all cases of their study, a correct response was received by the third answer, with the first one being correct 86.6% of the time. Having multiple crowdworkers discussing solutions together is more effective than having segregated workers provide separate answers. Lasecki et al. [106] provided workers with an interface where they can vote on existing answers as well as provide new answers. The user receives the most voted answer from the crowd-workers resulting in improved accuracy and speed. After receiving an initial description of an image, Zhong et al. [7] instructed workers to iteratively improve the description by increasing the amount of detail. As this approach resulted in an increase of the time cost, a parallelized approach would be more appropriate as workers can focus on describing different parts of the image [7]. We infer that to optimize between speed and quality, one should incorporate parallel descriptions and iterative improvements of each description.

The literature we reviewed in Section 5.2 provides other lessons one should incorporate when dealing with crowdsourcing for describing images to people with visual impairments. It is important to keep workers engaged and ready to provide descriptions to incoming queries. One could keep them engaged by instructing them to provide descriptions of previous query images or rate descriptions based on Table 5.4 [29]. Since sighted individuals can be employed as simulated visually impaired users [9], one could also have them rate descriptions without the associated image following the questions of Table 5.5. Finally, in describing images (as in other crowdsourcing tasks), it is generally helpful to have examples of good and bad descriptions, and to provide workers with interactive tutorials [7].

5.3.3 Photographic Issues

Many of the approaches we described in Section 5.2 required users to take photographs of certain scenes for which they had an inquiry. These studies uncovered the difficulties visually impaired users encountered while performing this task. Blind people have previously expressed their frustrations in not being capable of taking good quality pictures [29, 111]. For example, *VizWiz:LocateIt* [29] found that users have difficulty keeping the phone perpendicular to the ground. In addition, in trying to orient themselves to zoom into a specific view, they require constant guidance to the object, as we have found through our work in crossing intersections. Moreover, the studies revealed that visually impaired users have difficulties framing a view as they do not know how far they are from the entity, e.g., an object or a person. Brady et al. [30] evaluated the image quality based on blur, lighting, framing and composition (obscured objects), and found that only 15% of images taken by blind people were evaluated as perfect by sighted individuals, with an average of 1.6 errors per image, where an error, for example, was counted when the image was blurry or if the object was not contained in the frame or it was obscured, etc. This is not necessarily an issue when crowdsourcing is used as sighted people can still infer the correct answer [30], but can be detrimental for automatic systems.

An encouraging result is that after a one-year deployment of *VizWiz*, the authors discovered that users became better at taking pictures, with a reduction of 0.30 errors. Instead of individual images, a better approach would be to have a video stream with a sighted individual as it would allow the worker to instruct the user in real-time. This allows for sequential interaction, and was found to alleviate many of the stability issues by helping users properly orient the camera [106]. The problem with video streaming is the associated costs as we expect users would go through their data quota quickly. Image stitching can capture a larger view of the scene using only two frames while keeping data usage low. Zhong et al. [7] designed a user interface that guided users to pan the camera while it captured key frames for the stitching process, which used three to six frames. This was preferred by the blind participants as they felt it was easier and less stressful than following a sighted worker's instructions.

5.4 A Future Direction

Based on the literature we reviewed in this chapter, in future work, we would like to design a crowd-sourcing mobile system that provides blind individuals with visual descriptions of their surroundings. This application would draw inspiration from the approach of the *Autour* smartphone application, which renders the places around users spatially using *Google Places* or *Foursquare*, with Open Street Map for intersections and parks data. The proposed application would provide descriptions of various objects and people around the user, including information regarding their relationships and/or the actions they are performing. Upon receiving a new query image, we can instruct sighted crowdworkers to follow the guideline questions presented in Table 5.3. As other approaches have found, it would be beneficial to have workers collaborate while generating the description, which can be achieved through their user interface. To ensure they understand the requirements, we can provide an interactive tutorial as well as examples of good and bad descriptions through the interface, similar to previous approaches [7].

Such an application should maintain a pool of workers available to provide descriptions of new query images from blind users, as was done by previous methods [29]. This can be accomplished by having crowdworkers provide descriptions of previous query images. Workers can also be employed to rate the descriptions given the associated images by following the guidelines of Table 5.4. In addition, workers can be employed to play the role of simulated blind individuals by only showing them descriptions and an audio recording of the environment, and asking them to rate the description based on the guidelines of Table 5.5. To increase user engagement, we can incorporate concepts from gamification in crowdsourcing [112]. The *ESP game* proposed by Ahn et al. [113] explored labelling images in the context of an interactive and fun computer game. Ratings obtained from workers, with the role of a sighted or simulated blind worker, can be used to provide a score to the original description provider(s).

Acquiring images would be an important and challenging part of this application. As we have already explored in this chapter, previous approaches have required the user to point the camera of the phone towards the target entity for the query image. The most efficient and preferred approach was the use of image stitching employed by Zhong et al. [7]. Their method generated detailed descriptions as we could parallelize the workers' workload, with each one focusing on specific regions in the image to describe. *Aira* [105], a recently proposed paid service, connects users to specialized workers who are hired by the company, and assist the user with various visual tasks including micro-navigation, i.e., the task of assisting the user in navigating to a nearby location when the GPS is unreliable. The user can choose to use the application directly on their smartphone or using Horizon Smart Glasses equipped with a video camera for an extra cost of $25 (USD) per month. Instead, using an omnidirectional camera offers an opportunity to view the complete surrounding of the user. Following a similar framework as Zhong et al. [7] did, we can describe different areas of the 360-degree view and inform the user of the scene using spatialized audio rendering, similar to the approach by Blum et al. [35] employed in *Autour*. The social acceptability of this camera, and its positioning on the user, would have to be evaluated through user studies.

As Brady et al. [30] found, blind users have a higher-than-usual rate of abandonment due to usability issues. We expect that this rate could rise if the barrier for entry is too high due to high associated costs of using the application, e.g., the cost of the glasses or omnidirectional camera. As such, we believe it will be important to provide a second mode of interaction such as the image stitching approach of *RegionSpeak*. Other aspects of such an application, such as the gamification of the descriptions, the usability of the rating system for both the users and the workers (or volunteers), would have to be assessed through user studies.

The end goal of this application would be to release an image description dataset with the special property that descriptions are useful and adequate to blind individuals. This dataset would allow AI practitioners to build deep learning models that provide rich scene descriptions. This corpus would also contain information about how blind people rate descriptions, which could serve as data to train a model that automatically scores the usefulness of a description. We expect the release of such a dataset would have the same effect on the blind community as the release of the VizWiz dataset [45].

Chapter 6

Conclusion

The exploration of new environments is a challenging task for people living with visual impairments. Deep learning offers a way to analyze the world around the user using visual content. In this thesis, we explored the effectiveness of using deep learning in assistive technologies for the blind community. We focused on two main challenges associated with the exploration of outdoor environments: crossing intersections with minimal veering, and providing descriptions of scenes surrounding the user.

To understand how blind people cross intersections, we conducted observational studies where participants were asked to qualitatively explain the steps they follow. Using the lessons we learned, a dataset was collected from various intersections in urban areas of downtown Montreal, Canada. Using sighted individuals (task experts), this dataset was labelled with the correct heading that should be followed before and during crossing. An imitation learning agent was trained to learn a policy that maps from a camera's field of view to the correct heading to follow. We designed a mobile application that assists visually impaired users to cross intersections using the trained agent, with heading information provided through auditory feedback. To assess the application's effectiveness, we conducted a user study with eight blind participants. This initial study showed that the application significantly improved users' ability to locate and align themselves *faster* with the target crossing, and it did not affect their speed of crossing. Although not statistically significant, the study showed that the application also reduced the likelihood that a user veered outside the crossing lanes. To address some of the limitations we identified in the user study, we conducted a series of iterative experiments with a blind colleague. A final application was designed to address many of these limitations, employing a combination of on-device IMU sensors and the imitation learning agent.

To understand the challenges faced by visually impaired users with understanding their surrounding, we surveyed recent literature that provided users with descriptions of visual content, or answered questions regarding this content. The literature included methods that used automatically generated image descriptions as well as methods that employed crowdsourced workers. In Section 5.3, we summarize our findings by identifying key features required by this community. In order for descriptions to be useful to this community, it is important that they identify specific features of the image. In the context of exploring outdoor environments, we also present the lessons learned regarding the challenges faced by blind people in capturing images. For these descriptions to eventually be automatically generated, future work should design a mobile application that employs crowdsourced workers who follow the instructions we identified in Section 5.3.1 combined with the lessons we learned from crowdsourcing approaches, presented in Section 5.3.2, such as the proposed direction in Section 5.4. A successful application would result in the generation of a new general image-description dataset geared towards the visually impaired community.

The research we presented in this thesis suggests that deep learning has the potential to be an effective tool for assisting visually impaired users in various parts of outdoor exploration, and possibly with indoor scene understanding. The ability to have these models operate on mobile devices makes them suited for the assistive technology domain. For these models to be effective, it is important to have training data that reflects the needs of this user population. To obtain such data, designers should understand the potential user's capabilities and requirements by including members of the blind community in the collection process. The resulting models would provide information that adequately addresses their needs.

An important consideration for future designers is the fact that blind people place a great deal of trust in assistive technologies, suggested through our repeated trials with the same participants in Section 4.5, and was found from previous research in automatic image descriptions for alternative text [8, 9]. Unfortunately, as we have found through our research, deep learning models can fail at times in unexpected and incomprehensible ways. One should provide users with a level of confidence in the predictions to promote user skepticism by incorporating this information in the feedback. Bayesian Deep Learning [100] has recently been used in autonomous vehicle safety as it provides a measure of uncertainty in neural networks. We expect that this approach will be equally important in assisting blind people with autonomous outdoor exploration.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[3] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *arXiv preprint arXiv:1610.02391*, 2016.

[4] Google and Open Handset Alliance, "Android API Guide." `https://developer.android.com/guide/`. Accessed: 2018-10-05.

[5] "CaptionBot - Powered by Microsoft Cognitive services - Microsoft Corporation." `https://www.captionbot.ai/`. Accessed: 2018-10-11.

[6] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *European Conference on Computer Vision*, pp. 382–398, Springer, 2016.

[7] Y. Zhong, W. S. Lasecki, E. Brady, and J. P. Bigham, "Regionspeak: Quick comprehensive spatial descriptions of complex images for blind users," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2353–2362, ACM, 2015.

[8] H. MacLeod, C. L. Bennett, M. R. Morris, and E. Cutrell, "Understanding blind people's experiences with computer-generated captions of social media images," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 5988–5999, ACM, 2017.

[9] E. Salisbury, E. Kamar, and M. R. Morris, "Toward scalable social alt text: Conversational crowdsourcing as a tool for refining vision-to-language technology for the blind," *Proceedings of HCOMP 2017*, 2017.

[10] M. Diaz, R. Girgis, T. Fevens, and J. Cooperstock, "To veer or not to veer: Learning from experts how to stay within the crosswalk," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] O. Lahav, D. Schloerb, S. Kumar, and M. Srinivasan, "A virtual environment for people who are blind - a usability study," *Journal of Assistive Technologies*, vol. 6, no. 1, pp. 38–52, 2012.

[12] HumanWare, "Trekker Breeze+ handheld talking GPS." `https://store.humanware.com/hau/trekker-breeze-plus-handheld-talking-gps.html`. Accessed: 2018-09-06.

[13] S. A. Panëels, D. Varenne, J. R. Blum, and J. R. Cooperstock, "The walking straight mobile application: Helping the visually impaired avoid veering," in *Proceedings of ICAD13*, pp. 25–32, 2013.

[14] S. Panëels, A. Olmos, J. Blum, and J. R. Cooperstock, "Listen to it yourself! evaluating usability of "what's around me?" for the blind," in *Human Factors in Computing Systems (CHI)*, 2013.

[15] J. M. Loomis, R. L. Klatzky, and N. A. Giudice, "-sensory substitution of vision: Importance of perceptual and cognitive processing," in *Assistive technology for blindness and low vision*, pp. 179–210, CRC Press, 2012.

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

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[19] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[22] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*, pp. 2048–2057, 2015.

[23] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and vqa,” *arXiv preprint arXiv:1707.07998*, 2017.

[24] “Microsoft Corporation.” `https://www.microsoft.com/en-us/seeing-ai`. Accessed: 2018-10-10.

[25] “TapTapSeeApp Inc..” `https://taptapseeapp.com/`. Accessed: 2018-10-11.

[26] “BlindSight By *Neuro X Labs*.”

[27] “OrCam Technologies Ltd..” `https://www.orcam.com/en/myeye2/`. Accessed: 2018-10-10.

[28] “Eyra Ltd..” `https://horus.tech/?l=en_us`. Accessed: 2018-10-10.

[29] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, *et al.*, “Vizwiz: nearly real-time answers to visual questions,” in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 333–342, ACM, 2010.

[30] E. Brady, M. R. Morris, Y. Zhong, S. White, and J. P. Bigham, “Visual challenges in the everyday lives of blind people,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2117–2126, ACM, 2013.

[31] H. Shen, K.-Y. Chan, J. Coughlan, and J. Brabyn, “A mobile phone system to find crosswalks for visually impaired pedestrians,” *Technology and disability*, vol. 20, no. 3, pp. 217–224, 2008.

[32] J. L. Souman, I. Frissen, M. N. Sreenivasa, and M. O. Ernst, “Walking straight into circles,” *Current Biology*, vol. 19, no. 18, pp. 1538–1542, 2009.

[33] D. Guth, “Why does training reduce blind pedestrians veering,” *Blindness and brain plasticity in navigation and object perception*, pp. 353–365, 2007.

[34] D. A. Guth, E. W. Hill, and J. J. Rieser, “Tests of blind pedestrians’ use of traffic sounds for street-crossing alignment.,” *Journal of Visual Impairment & Blindness*, 1989.

[35] J. Blum, M. Bouchard, and J. R. Cooperstock, "What's around me? spatialized audio augmented reality for blind users with a smartphone," in *Mobile and Ubiquitous Systems (Mobiquitous) - Best Paper Award*, Springer, Dec. 2011.

[36] "BlindSquare." `http://www.blindsquare.com/`. Accessed: 2018-10-10.

[37] J. R. Blum, D. G. Greencorn, and J. R. Cooperstock, "Smartphone sensor reliability for augmented reality applications," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pp. 127–138, Springer, 2012.

[38] "Google AI Blog - Using Global Localization to Improve Navigation." `https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html`. Accessed: 2019/02/20.

[39] "Be My Eyes - Mobile Application." `https://www.bemyeyes.com/`. Accessed: 2018-10-11.

[40] "CloudSight Inc.." `https://cloudsight.ai/`. Accessed: 2018-10-11.

[41] Jeremy R. Cooperstock - McGill University, "Autour - Mobile Application." `http://autour.mcgill.ca/en/`. Accessed: 2018-10-11.

[42] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional gan," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2989–2998, IEEE, 2017.

[43] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, and B. Schiele, "Speaking the same language: Matching machine to human captions by adversarial training," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[44] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollr, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv:1504.00325*, 2015.

[45] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," *arXiv preprint arXiv:1802.08218*, 2018.

[46] "VizWiz ECCV Workshop." `http://vizwiz.org/workshop/`. Accessed: 2018-10-11.

[47] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318, Association for Computational Linguistics, 2002.

[48] A. Scott, J. Barlow, B. Bentzen, T. Bond, and D. Gubbe, “Accessible pedestrian signals at complex intersections: Effects on blind pedestrians,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2073, pp. 94–103, 2008.

[49] C. N. I. for the Blind, “Cnib: Position for accessible pedestrian signals in canada,” 2003.

[50] New York City Department of Transportation, “Accessible Pedestrian Signals Program Status Report.” `http://www.nyc.gov/html/dot/downloads/pdf/2015-aps-program-status-report.pdf`, 2015. Accessed: 2017-06-30.

[51] D. Fernandes, *Vehicle-pedestrian Accidents at Signalized Intersections in Montreal*. PhD thesis, McGill University, 2013.

[52] M.-M. R. Centre, “List of audible pedestrian signals installed and to come,” 2013.

[53] D. A. Ross and B. B. Blasch, “Wearable interfaces for orientation and wayfinding,” in *Proceedings of the fourth international ACM conference on Assistive technologies*, pp. 193–200, ACM, 2000.

[54] N. Mohssen, R. Momtaz, H. Aly, and M. Youssef, “It’s the human that matters: accurate user orientation estimation for mobile computing applications,” in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp. 70–79, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[55] A. Arditi, J. D. Holtzman, and S. M. Kosslyn, “Mental imagery and sensory experience in congenital blindness.,” *Neuropsychologia*, vol. 26 1, pp. 1–12, 1988.

[56] D. Ahmetovic, C. Bernareggi, A. Gerino, and S. Mascetti, “Zebrarecognizer: Efficient and precise localization of pedestrian crossings,” in *ICPR*, 2014.

[57] D. Ahmetovic, R. Manduchi, J. M. Coughlan, and S. Mascetti, “Mind your crossings: Mining gis imagery for crosswalk localization,” *ACM Transactions on Accessible Computing (TACCESS)*, vol. 9, no. 4, p. 11, 2017.

[58] V. Ivanchenko, J. Coughlan, and H. Shen, “Staying in the crosswalk: A system for guiding visually impaired pedestrians at traffic intersections,” *Assistive technology research series*, vol. 25, no. 2009, p. 69, 2009.

[59] M. Poggi, L. Nanni, and S. Mattoccia, “Crosswalk recognition through point-cloud processing and deep-learning suited to a wearable mobility aid for the visually impaired,” in *International Conference on Image Analysis and Processing*, pp. 282–289, Springer, 2015.

[60] M. Poggi and S. Mattoccia, "A wearable mobility aid for the visually impaired based on embedded 3d vision and deep learning," in *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pp. 208–213, IEEE, 2016.

[61] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[63] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 71–78, Association for Computational Linguistics, 2003.

[64] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

[65] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

[66] D. Elliott and F. Keller, "Comparing automatic evaluation measures for image description," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 452–457, 2014.

[67] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1473–1482, 2015.

[68] K. Tran, X. He, L. Zhang, J. Sun, C. Carapcea, C. Thrasher, C. Buehler, and C. Sienkiewicz, "Rich image captioning in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 49–56, 2016.

[69] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[70] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell, "Language models for image captioning: The quirks and what works," *arXiv preprint arXiv:1505.01809*, 2015.

[71] S. Aditya, Y. Yang, C. Baral, C. Fermuller, and Y. Aloimonos, "From images to sentences through scene description graphs using commonsense reasoning and knowledge," *arXiv preprint arXiv:1511.03292*, 2015.

[72] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 1798–1828, Aug. 2013.

[73] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[74] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[75] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[76] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 21, 2017.

[77] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[78] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[79] B. D. Argall, "Human autonomy through robotics autonomy." `https://youtu.be/od6V1tt0ctc`, 2016.

[80] D. A. Pomerleau, "Advances in neural information processing systems 1," ch. ALVINN: An Autonomous Land Vehicle in a Neural Network, pp. 305–313, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989.

[81] J. Togelius, R. D. Nardi, and S. M. Lucas, "Towards automatic personalised content creation for racing games," in *2007 IEEE Symposium on Computational Intelligence and Games*, pp. 252–259, April 2007.

[82] S. Ross and J. A. Bagnell, "Efficient reductions for imitation learning," in *In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS*, 2010.

[83] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016.

[84] D. K. Kim and T. Chen, "Deep neural network for real-time autonomous indoor navigation," *arXiv preprint arXiv:1511.04668*, 2015.

[85] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[86] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, (USA), pp. 1097–1105, Curran Associates Inc., 2012.

[87] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision (IJCV)*, vol. 88, pp. 303–338, June 2010.

[88] J. Yosinski, J. Clune, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *In ICML Workshop on Deep Learning*, 2015.

[89] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[90] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, Oct 2010.

[91] A. Giusti, J. Guzzi, D. C. Cireşan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro, *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2016.

[92] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *CoRR*, vol. abs/1605.07678, 2016.

[93] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, June 2016.

[94] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “Enet: A deep neural network architecture for real-time semantic segmentation,” *CoRR*, vol. abs/1606.02147, 2016.

[95] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pp. 807–814, 2010.

[96] T. Tieleman and G. Hinton, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude.” COURSERA: Neural Networks for Machine Learning, 2012.

[97] D. Clark-Carter, A. Heyes, and C. Howarth, “The efficiency and walking speed of visually impaired people,” *Ergonomics*, vol. 29, no. 6, pp. 779–789, 1986.

[98] A. Guide, “Effects of walk signal characteristics,” 2005.

[99] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” *arXiv preprint arXiv:1712.07107*, 2017.

[100] R. M. Neal, *Bayesian learning for neural networks*, vol. 118. Springer Science & Business Media, 2012.

[101] E. Frank and M. Hall, “A simple approach to ordinal classification,” in *European Conference on Machine Learning*, pp. 145–156, Springer, 2001.

[102] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv preprint arXiv:1706.05098*, 2017.

[103] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[104] N. Radwan, A. Valada, and W. Burgard, “Multimodal interaction-aware motion prediction for autonomous street crossing,” *arXiv preprint arXiv:1808.06887*, 2018.

[105] “Aira Tech Corp..” `https://aira.io/`. Accessed: 2018-11-07.

[106] W. S. Lasecki, P. Thiha, Y. Zhong, E. Brady, and J. P. Bigham, “Answering visual questions with conversational crowd assistants,” in *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, p. 18, ACM, 2013.

[107] "BlindTool A mobile app that gives a sense of vision to the blind with deep learning, 2015." `https://github.com/ieee8023/blindtool`. Accessed: 2018-10-26.

[108] S. Wu, J. Wieland, O. Farivar, and J. Schiller, "Automatic alt-text: Computer-generated image descriptions for blind users on a social network service.," in *CSCW*, pp. 1180–1192, 2017.

[109] D. Prelec, "A bayesian truth serum for subjective data," *science*, vol. 306, no. 5695, pp. 462–466, 2004.

[110] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft COCO captions: Data collection and evaluation server," *CoRR*, vol. abs/1504.00325, 2015.

[111] C. Jayant, H. Ji, S. White, and J. P. Bigham, "Supporting blind photography," in *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pp. 203–210, ACM, 2011.

[112] B. Morschheuser, J. Hamari, and J. Koivisto, "Gamification in crowdsourcing: a review," in *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pp. 4375–4384, IEEE, 2016.

[113] L. Von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 319–326, ACM, 2004.