A Transfer Learning Approach to Surface Detection for Accessible Routing for Wheelchair Users

Valeria Mokrenko*, Haoxiang Yu*, Vaskar Raychoudhury*, Janick Edinger†, Roger O. Smith‡, Md Osman Gani§
* Department of Computer Science & Software Engineering Miami University, Oxford, OH, USA
† Department of Informatics Distributed Operating Systems, University of Hamburg, Hamburg, Germany
‡ Department of Occupational Science and Technology, University of Wisconsin, Milwaukee
§ Department of Information Systems, University of Maryland Baltimore County

***Abstract*—The nature of the surface has a significant effect on how wheelchair users experience locomotion. The preferred surfaces for wheeled mobility must be even, firm and smooth while generating adequate friction. The development of accessible road maps that include ground conditions is therefore of utmost importance. Our prior work has shown how such maps can be created using surface-induced vibration data collected by motion sensors embedded in smartphones and then classifying them with machine learning algorithms. To make data collection scalable, participatory crowd-sensing can be used, where users collect and transmit sensor data while traveling on wheelchairs. The complexity here is that wheelchairs widely vary in type (manual, power-assist, power), weight, number and nature of wheels, therefore the sensor data generated by different wheelchairs varies greatly. Collecting training data on each individual wheelchair type to develop classification models is not feasible. To address this problem, in this paper we explore the possibility of transferring knowledge from known wheelchairs to unknown types. We develop a transfer learning algorithm to classify 15 surfaces with minimal training data from different wheelchairs. Our experiments with 47 subjects show that surface classification knowledge, learned from sensor data generated by manual wheelchairs, can be transferred to a power wheelchair with up to 90.02% accuracy. This allows crowd-sensing to be used effectively for data collection for generating accessible route maps. We integrate our transfer learning approach into our system for accessible routing, which we developed in previous work.**

***Index Terms*—accessible routing, manual wheelchair, power wheelchair, transfer learning, surface classification**

I. Introduction

Accessible routing and navigation allows wheelchair users to freely roam around unknown terrains. Routes which are accessible should not contain any temporary or permanent barriers to wheeled mobility, such as, narrow sidewalks, stairs, steep slopes, broken surfaces, or sudden curb drops. Existing research on accessible routing and navigation systems [1], [2] aim to provide multiple routes to a user following the *Americans with Disabilities Act* (ADA) standards of safety criteria that building landmarks, sidewalks and streets need to have for accessibility [3]. Since outdoor accessibility can change over time or due to adverse weather conditions, information on a path may not always be available or up-to-date in government or city records. To address this dearth of real time information, crowd-sourcing has been utilized to detect and identify the barriers of accessibility in public areas [4], [5]. In addition, data collected using various inertial sensors (accelerometers and gyroscopes) [6], [7], [8], [9] use GPS and GIS for generation of accessible routes.

TABLE I: Relevant context dimensions for accessible routing for wheelchair users. The highlighted cells indicate which dimensions we have considered in the present work.

Wheelchair	User	Environment
Type	Disability	Type
Material	Degree of disability	Time
Number of Wheels	Age	Weather
Drive Type	Weight	Obstacles
Seat to floor height	Height	Width of surface
Weight capacity	Gender	Type of surface
Seat width	Purpose of travel	
Overall width		

We have extensively studied the problem and developed , an inertial sensor-based accessible routing and navigation system that works by classifying surfaces using machine learning algorithms [10]. *Wheelshare* was the first system that empirically studied the effect of surface-induced vibrations collecting and classifying data from 32 different indoor and outdoor surfaces found across Austria, China, France, Germany, India, and the USA [11], [10]. Our system achieved an accuracy of up to 97% using accelerometer and gyroscope data and an accuracy of 92.3% using only accelerometer data all available through smartphone embedded sensors.

However, while we tried to port our results to a power wheelchair, we could not find any direct method to achieve that objective. A power wheelchair is significantly heavier than a manual wheelchair as it has a motor and a battery and shorter but wider wheels with pneumatic, solid or flat-free tires. Moreover, power wheelchairs mostly have two pairs of caster wheels. Therefore, they produce different vibration patterns than the manual wheelchairs. Also, the machine learning algorithms which successfully classified different surfaces using manual wheelchair data, could not be applied to the new data collected using power wheelchair without first training them. We have extensively analyzed and then identified the relevant parameters which are associated with three dimensions of accessible routing - *user*, *wheelchair* and the *environment*. Table I shows the relevant context dimensions of accessible routing which need to be distinguished to make accurate predictions. We realized that given the wide variety of built environments as well as natural areas, types of wheelchairs, differences in user abilities and endurance levels, and weather conditions, it is nearly impossible to test every combination of the parameters using individual experiments or to train machine learning classifiers separately every time. We found no existing methods that can transfer their knowledge of accessibility and use data collected from one-type of sensor-augmented wheelchair for a particular environment to predict the accessibility of unknown paths while using other types of wheelchairs. Similarly, current machine learning algorithms work in isolation and fail to achieve cross-domain knowledge transfer. They require data to be acquired for training in order to make predictions about the features present in the built environment. In contrast, transfer learning uses existing knowledge on performing a task in one domain to help perform a somewhat different task in a different domain. Since models often require substantial data in order to solve problems, transfer learning utilizes knowledge such as features and weights from a previously trained model to a new task.

Given these challenges, we intended to investigate the following research question in relation to developing accessible routing solutions for wheelchair users: *What is the viability of knowledge transfer from one particular type of mobility aid device (e.g., manual wheelchair) to other types (e.g., power wheelchair, walker, or mobility scooter)?*

While looking for solutions to the aforementioned research questions and to achieve rapid prototyping of accessible routing solutions across multiple domains, in this paper we propose the first ever transfer learning based surface detection approach for accessible routing named *WheelTransfer*. WheelTransfer can optimize the learning performance and accelerate training of the model to make accurate route predictions. WheelTransfer makes the following novel contributions:

- We design a transfer learning based surface classification system for accessible route generation. We have successfully demonstrated knowledge transfer from a manual wheelchair to a power wheelchair up to 90.02% accuracy using only a minimal amount of new training data. This opens up a new vista of research in accessible routing and has the potential to shape the future research directions.
- Experiments using 13 standard indoor and outdoor surfaces and 2 curbs across the USA and China using 3 different manual and power wheelchairs and 47 participants give evidences that our proposed system is scalable and adaptable and it propagates the knowledge of automated and objective surface classification across multiple different domains.
- We integrate transfer learning in our existing environment for accessible routing and, thus, make data collection for this system cheaper and scalable.

II. Related Works

In this section we discuss the relevant state-of-the-art in accessible routing as well as in transfer learning.

A. Accessible Routing and Navigation Systems

Following are the different kinds of accessible routing and navigation systems for wheelchair users that aim to consider the knowledge of the built environment and the needs of the users for route generation.

1) Mobility assistants: Initial systems were mobility assistants that focused on simply creating GIS models for users and which were tested in one small area. MAGUS [12], [13] was among the first such systems and it assigned an impedance score using network analysis tools for different route recommendations based on minimum barriers, shortest distance, fewest slopes and tough surface terrain as well as limiting the number of road crossings. One of the first applications of machine learning to accessibility routing came from Karimanzira et al. [14]. Another system, U-Access [15], considers many different types of environmental objects that can affect which routes are chosen depending on the physical ability of the user. Mobility assistants would often use GIS and GPS software [16] to create a network-based accessible map which would have clear barriers and accessible points of interest (POI) such as facilities or stairs. Several systems [6], [7], [8], [9] require the use of smartphone gyroscope, accelerometer and location information about the user and their travel experience.

2) Crowd-sourced recommendation systems: 'Crowdsourcing' [4], [5] refers to the "people's knowledge" and is provided by a community of wheelchair users regarding barriers, obstacles and POIs both indoors and outdoors. The RouteCheckr system [17] enables collaborative data annotation and creates personalized routes for users. Mobile Pervasive Accessibility Social Sensing (mPASS) [18] is a system that creates personalized routing paths based on data gathered from crowdsourcing as well as sensory data from geo-referencing. One of the most up-to-date, collaborative providers of crowd-sourced data is OpenStreetMap [19], [20], whose data was incorporated in route calculation for systems such as eNav [21].

3) Defining Accessibility Criteria: The challenge with modeling accessible routes comes from the unpredictability of the objects that can affect accessibility. Routes may deem uncomfortable for travel if they are temporarily flooded, too steep, have an uneven surface, have a slope beyond one-in-twenty or have a path narrower than 915mm [3]. One algorithm builds a routing network based on multiple criteria on accessibility gathered from OpenStreetMap (OSM) data [22]. Kawabata [23] proposes context-adaptable pedestrian navigation system, which creates user preferred routes by considering stairs and road surface but not slope. Rahaman et al [24] proposed a network graph that does consider slope, and use a multi-objective A* search algorithm (CAPRA) [25] that reports distance and elevation. Sahelgozin et al created the optimization method, Ubiquitous Pedestrian Way Finding Service (UPWFS) [26], and defined four criteria that needs to be met for a personalized path that suits users: length, safety, difficulty and attraction.

B. Transfer Learning Approaches

Transfer learning is the improved rate of learning a new task through existing knowledge from a related, already learned task. Transfer learning utilizes knowledge such as features and weights from a previously trained model. If labeled data exists for both the target and source domain, the model can solve the source task and be applied to a new problem in the target domain called multi-task learning. One form of transfer learning, referred to as domain adaptation, assumes that the new task is the same as the old task, but the input distribution is different [27]. Several approaches exist for transfer learning: *Feature-representation Transfer* requires identifying specific features or "denominators" from learning tasks and how would their meaning assist in understanding the relation between the source and target features. Its applications include face recognition applications with underrepresented data [28]. *Instance Transfer* identifies important instances from source data through boost instance weighing [29] or elimination [30]. *Model Transfer* requires that an accurate, predictive model from the source domain is trained, which can be then adapted to solve the target task given the data from the target domain. This approach is useful when considering storage capacity or applications when the original data is no longer needed [31]. Transferring features from a new domain to initialize an existing, trained network can increase the generalization performance to predict labels on the new target data [32].

III. WheelTransfer System Architecture

We integrate WheelTransfer into our initial *Wheelshare* system [11], [10] and present it as a comprehensive, consistent, efficient and accurate accessible routing solution. The system architecture is shown in Figure 1 and is composed of several sub-systems distributed between client and server side operations. Functionally, the operations of the system are divided into a training phase and a live phase. In the training phase, experiments are conducted both indoors and outdoors for curbs, flat paths and other common surface types. The surface type and slope data are used for classification model training. The live phase consists of a unobtrusive, participatory crowd-sensing technique to collect location-tagged path information from urban areas in order to identify accessible features and to recommend personalized routes in the built environment.

As shown in Figure 1, in the training phase, a registered wheelchair user can contribute to data collection (step 0 and step 1) using a smartphone (Android or iOS) embedded with motion sensors. Collected data is subject to a surface classification operation in the central server using machine learning (step 2). Once data is accumulated from the user's surroundings, the graphical overlay map is constructed with important path features (step 3) which helps to generate accessible routes (step 4) based on user initiated queries (step 5). The accessible route query and response for the user can take place during the testing phase with enough data to populate the server. Below, we describe different functional modules in more details.

A. Client side operations

Client side operations of the WheelTransfer system has two modules - a *data collection module* and a *routing and navigation module*. Both the modules are incorporated as parts of an Android application.

Data collection module (Step 1): The data collection module of the app allows user to register at the time of first usage and collects user information with a view to personalize the accessible routes. User name, height, weight, physical ability and type of mobility device (if any) is collected. Physical ability or impairment level of the user can be inserted based on some standard scale, such as American Spinal Injury Association (ASIA) Impairment Scale for spinal cord injuries/ diseases [33] or Expanded Disability Status Scale (EDSS) for multiple sclerosis [34]. The inertial sensors in the smartphone or the wearable device begins to collect the vibration data and sends as an HTTP POST request to our web server. The server sends an automatically identified surface type to the user who has the choice to correct the surface type if an erroneous classification is provided. The surface type and sensor data are then uploaded to the server database.

Routing and navigation module (Step 5): The user enters source and destination location coordinates to initiate a route request. The server returns multiple accessible routes with different path features which are then personalized based on user preferences such as acceptable curb heights or slopes. Also, a textual navigation pane is provided for the chosen route which gives turn-by-turn navigational instructions with distance, time and surface type information to cover each section of the route.

B. Server side operations

Server side operations of the WheelTransfer system has three modules for surface classification, feature identification, and route generation.

Surface Classification (Step 2) and Feature Identification (Step 3) Modules: In this module, the server uses machine learning algorithms to classify surfaces based on inertial sensor data. In the present work, we have especially extended the surface classification module (step 2) and successfully incorporated a transfer learning algorithm to apply the knowledge of the classifier. The original classifier is trained using manual wheelchair data to classify surfaces using only a minimal amount of power wheelchair data. User contributed surface vibration data (Figure 2) is processed and several accessible features and POIs are extracted and uploaded to the open source mapping platform called OpenStreetMap (OSM) [19]. More details of our surface classification method will be described in Section IV.

Route Generation Module (Step 4): The routing module uses open source information from OSM to gather data about the surrounding of a querying user. OSM allows geo-coding of location names and has the ability to tag information on nodes and ways for route planning. User location information collected using a GPS unit is labeled and attributed before being uploaded for route surveying. Different features either

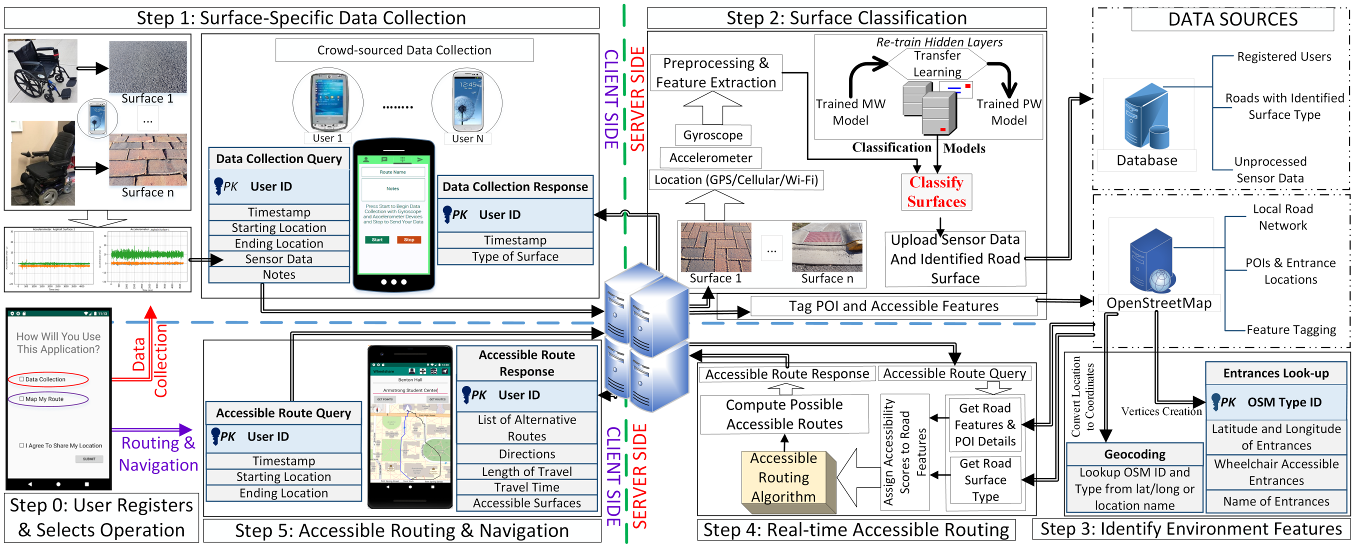

Fig. 1: Complete WheelTransfer System Architecture. Data collection and navigation is performed through a mobile application on the client side (steps 1 & 5). The server runs the surface classification and identifies features in the environments such as entrances into buildings (steps 2 & 3). Further it responds to routing requests from the application (step 4).

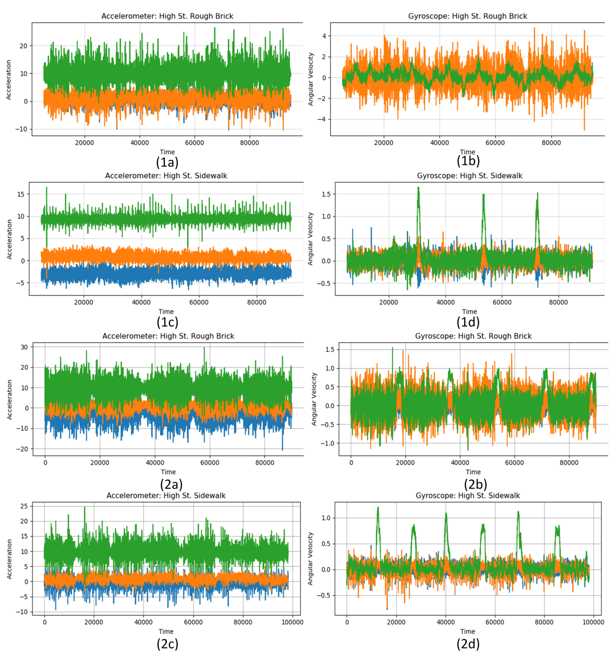

Fig. 2: Surface-induced vibrations caused by manual wheelchairs: (1a) acclerometer data and (1b) gyroscope data for High St. Rough Brick Surface; (1c) acclerometer data and (1d) gyroscope data for High St. Sidewalk surface. (2a-2d) Vibrations for the same surfaces caused by power wheelchair.

existing in OSM or by the machine learning classifier are used for multiple route recommendation for users with different mobility devices. Once the user chooses one of the many routes to navigate through, they can also decide to contribute their vibration data to further enrich the system. The surface data of the path is tagged and uploaded to the OSM after filtering.

IV. Knowledge Transfer From Manual to Power Wheelchair

Machine learning algorithms are used in order to understand the structure of the data in a particular domain of knowledge. The challenge is how to use previous knowledge gained from the abundance of information provided by manual wheelchair contributors in order to gain knowledge about the environment and create personalized routes for a user with a 'similar' mode of transportation. We know that there are factors of variation from the features extracted from vibration data. However, there may be generic commonalities that are both shared in the manual wheelchair (MW) and power wheelchair (PW) domains. In order to investigate that, once we trained machine learning models with the MW dataset which is extensive in number of participants (47), number of different surfaces (32), number of different wheelchairs (5) and number of smartphones (6) we planned to re-train the model with a handful of PW data. However, due to the scarcity of labeled data collected by power wheelchairs, reconstructing a new target model may not lead to efficient or reliable performance, especially in deep learning models that require tens of thousands of new samples in order to tune the parameters. To overcome this problem, we decided to use transfer learning which aims to transform the distributions from features trained by one model from the source domain to another model in the target domain. Transfer learning can lead to more scalable route calculation and reduced computational complexity, since data from manual wheelchairs does not need to be used after the initial training of the model. Even efficient algorithms such as SVM and NB will have a high-cost for training time and space as the surface knowledge increases. In order to make predictions about surfaces for users who are using a power wheelchair, we consider the problem as being in the target feature space and having different distributions than models trained on MW experiments.

A. Selecting ANN Model for Transfer Learning

Our application of machine learning models on MW data involves the creation of eight classifiers: K-Nearest Neighbor (KNN), Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVM), Naive Bayes (NB) and three deep learning models which are Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM). Among all of the selected models, RF had the highest classification accuracy at 89.59%. ANN had the highest accuracy among the deep learning models at 73.81% followed distantly by LSTM at 64.39% and CNN at 63.48%. The computational complexity of RF is $T*N*log(N)$, where T is the number of trees in the ensemble and N is the number of samples. When training is performed using the surfaces collected from MW experiments, we found that RF took about 150 times longer to train than DT, NB and KNN and about 3 times longer than SVM. In comparison, ANN training is about 10 times faster than RF, averaging 8 seconds, 2.4 times faster than CNN and 22.5 times faster than LSTM.

After training for 30 epochs, we noted that the training loss and test loss performed very well after just training for 10 epochs, remaining stable at 1.0. Stabilized performance at 25 epochs for CNN showed signs of underfitting, with train loss being slightly higher than test loss. LSTM appeared to show overfitting as testing data being slightly higher than training data at performance stabilizing at 23 epochs. Therefore, the ANN model is chosen for transfer learning as not only did it have the highest classification accuracy among the deep learning models, but also requires one of the fastest training times and performed well with the lowest epoch training.

B. Transfer Learning Algorithm

The transfer learning approach we chose is based on domain adaptation by fixing or "freezing" the weights on certain layers on the network that were trained on the original source data and training the unfixed layers on the new target data.

ANN model, which consists of 7 layers - an input layer, four hidden layers using the Rectified Linear Unit (ReLU) activation function, one hidden layer with batch normalization and an output layer with Softmax activation. The inputs are a trained deep learning model, labeled PW samples, and the number of layers desired to freeze in the trained model f, which ranges from 0 to 5. By setting each layer as not trainable, the weights are not updated when the models are retrained on the PW data. This method may increase the accuracy if the specific data from the source MW domain generalizes well on data from a target PW distribution. When $f = 0$, all of the weights for the layers are adapted to the new target domain, while when $f = 5$, none of the layers are adapted except the output layer and the hidden layers function. The model is then recompiled using the same Adam optimizer and loss function and fit on the PW data.

C. Classification Training Procedure

The goal of supervised training is to automatically detect a class label based on decision rules previously learned through known input patterns in training data. The model development is split into two main phases: *Source Domain Learning* (SDL) which creates two models each trained on either unsampled or resampled data, and *Target Domain Learning* (TDL) which uses an already trained artificial neural network M and applies the SLT technique given a specific number of fixed layers f in order to produce a model on the target domain. Both of the SDL models undergo TDL and are adapted to the target PW domain. The models from SDL and TDL are validated on test data from their corresponding learned domain, either MW or PW. The procedures *SDL* and *TDL* are explained in the following subsections.

1) Source Domain Learning: The raw data input from manual wheelchair sensors D_S undergoes data preprocessing, feature extraction, normalization and resampling before training. The pre-processing stage involves identifying and segmenting irregular vibrations or a change in speed that occurs from the initial speed up and slow down when participants begin and end the experiment. Since we are interested in the variation of movement and capturing the high and low peaks in the accelerometer and gyroscope data, we avoid any smoothing procedure that may cause false surface predictions. In the feature extraction stage, we used a windows-based approach with a window size of two seconds or 100 data points for acceleration and gyroscope traces to compute 22 commonly used features. Since four different phones were used for our data collection (not more than 2 at a time), the values greatly depend on the lower and upper bound of the frequency range. Not only do the phones have different sampling frequency, they also have different types of sensors with varying make and qualities introducing noises. Therefore, these errors are most commonly addressed by scaling the norm of the individual sensors by normalizing after extracting the features. The data from the sensors of each phone are transformed individually using a StandardScaler for each phone type, which subtracts the mean value for each data column and divides by the standard deviation. Then, similar surface features are grouped into categories shown in Table IIa formed using the method described in Section 5.3. We used 20% of the data for testing the models. Since there is an uneven amount of instances for each surface type, it is necessary to resample the feature data in the training set. We perform oversampling to generate synthetic known class instances for each category, then undersample the majority classes using Edited Nearest Neighbor (ENN) technique. We then have two versions of data, unsampled and resampled, which is used as input for training our models. We initialize a dictionary for each version of the training data in order to store all the trained models and their accuracy results on the test data. We then train 8 different models on each dataset: Decision Tree (DT), Random Forest (RF), K-nearest neighbor (KNN), Naive Bayes (NB), Support Vector Machines (SVM), Artificial Neural Networks (ANN), Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). Validation includes calculating the accuracy, precision, recall and F1-score of each model. The model and its accuracy are stored in the dictionary.

2) Target Domain Learning: The raw data from power wheelchair sensors is processed in a similar way as the manual wheelchair sensor data. Data segmentation, feature extraction and normalization is performed on the data. Training experiments which involved surface groups S2, S3 and S4, reduced the number of possible classes for prediction. We select the number of layers to fix in the ANN model trained on MW data and use the SLT procedure to apply transfer learning. We validate the retrained model to the test data from the PW domain. The average scores for 10 different validations in the TDL procedure are listed in Table VI and explained in Section 6. Based on previously learned knowledge about MW domain, the re-trained knowledge layers can predict surface types of previously unseen sensor data in the target PW domain.

V. Surface Data Collection Experiments

Once our transfer learning model is ready, we started training it with the manual and power wheelchair data. Data collection using the manual wheelchair occurred on 14 days between December 2018 to January 2020. A total number of 47 participants contributed data from 32 different surfaces in USA, India, China, Germany, Austria and France using iPhone X, Samsung Galaxy S9+, Samsung Galaxy S7 edge, and Samsung Galaxy J7 and a wearable sensing device that was attached to 5 different manual wheelchairs. However, in this paper, we mainly focus on using transfer learning to transfer the knowledge of manual wheelchair (first type of mobility-aid device) based service classification technique and apply it to the domain of power wheelchair (second type of mobility-aid device).

A. Device Selection

1) Primary Device Selection: Manual wheelchairs were selected as the first type of mobility-aid device due to their light weight, increased availability and popularity. They usually weigh about 40 lbs, has short-term or temporary use and has limited customization or adjustment. The device has a maximum weight capacity of 350 lbs and has solid rubber type wheels. Over 400 experiments were performed on surface data collection using this device. More details about our data collection procedure and experiments can be found at [11], [10].

2) Secondary Device Selection: The decision to choose a power wheelchair as the secondary device resulted from its similar popularity in the United States among middle-aged wheelchair users [35]. An automatic or power wheelchair is equipped with a motor and batteries usually underneath the seat. Current automatic wheelchairs allow for custom speed setting at the cost of more battery consumption. The selected power wheelchair weighs 260 lbs, uses pressurized wheels and has a maximum weight capacity of 265 lbs. It has a maximum turning radius of 25 inches, can go up to 5 mph and a slope capacity of 6 degrees.

B. Performing Data Collection

For manual wheelchair data collection, 22 participants in USA and 6 participants from China (Jiangsu Province) were used. For power wheelchair experiments, data collection occurred for a total of 19 participants. Data collection in USA were in the campus of Miami University and in the town of Oxford, OH, USA. More details about the age and body weight of participants are listed in Table III. This project has necessary IRB approval with Protocol ID# 01738r. The placement of phones at various positions allows for a better understanding of how sensor data differs according to their placement on the wheelchair. The recordings may also be affected by the proximity of the phone to the ground and the direction it faces. Each participant is required to continuously move the wheelchair on a specific surface for three minutes while the gyroscope and accelerometer sensors are continuously collecting the vibration information. The only exception to the three minute duration is on curbs, where the time taken is from the point where the participant starts to move forward to the time where the participant stops at the bottom or at the top of the curb height completely. We only chose curbs with ramps to ensure participant safety. Figure 3 shows the different types of surfaces that were used for data collection for manual wheelchair experiments. We have considered the various curbs in Figure 3 (ix-xi) as separate surfaces based on the user position and direction of travel. So, going up the curb or "Up Curb" produces different vibration than going down the curb or "Down curb". In total there are 13 different surfaces (i) to (viii) and (xii) to (xvi) and two curb features, and they comprise 15 different surface groups (A to O) as listed in Table IIa.

C. Data Pre-Processing

In order to regroup the 15 different surface categories into even smaller number of categories based on identical vibration patters, we performed a Welch T-Test for unequal variances for each small iteration of time for surface data from the Samsung S7 phone, which was used to collect the most data. The test is performed 35 times using approximately 2 seconds of data for all surface pairings except for curbs for a total of 70 seconds of data. From the results reported in Table IV, the average P-value is greater than 0.05 for all groups, therefore there is not enough evidence to reject the null hypothesis that the population mean and standard deviation for the two surfaces are equal using the two tailed t-test. Grouping would lessen the complexity of multi-class classification and may lead to more

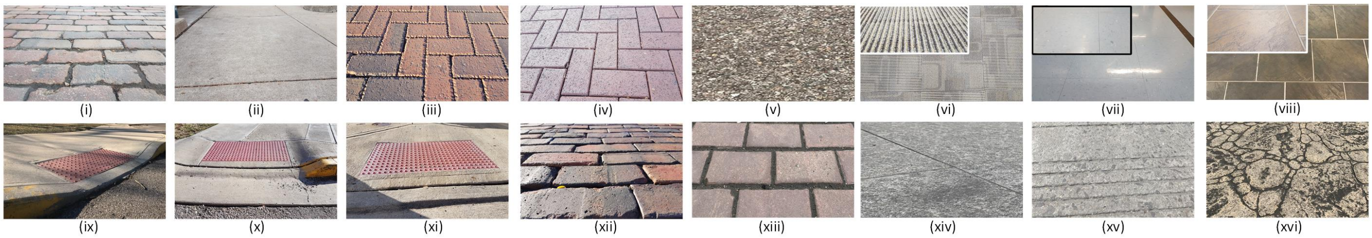

Fig. 3: Data from Various Surface Types Collected in Oxford, OH and China: **Surfaces Considered in USA:** (i) High St. Brick, (ii) High St. Sidewalk, (iii) High St. Smooth Brick 1 (with rough edges), (iv) High St. Smooth Brick 2, (v) Chestnut St. Parking Asphalt and Marcum Parking lot, (vi) Benton *Indoor* Carpet, (vii) Benton *Indoor* Mat Surface, (viii) Benton *Indoor* Tiles, (ix)-(xi) Various Curbs at E Park Pl and Marcum Conference, (xii) High St. Rough Brick **Surfaces Considered in China:** (xiii) Gusu Brick, (xiv) Lu Shan Lu Granite Tiles, (xv) Lu Shan Lu Asphalt Sidewalk (with ridges), (xvi) Gusu Patterned Flagstone Road

(a) TABLE II(a): The original 15 categories for surfaces (S1) merged into 14 (S2), 13 (S3) and 12 categories (S4).

Surfaces		S1	S2	S3	S4
1	High St. Brick	A	A	A	A
2	High St. Sidewalk	B	B	B	B
3	High St. Smooth Brick 1	C	C	C	C
4	High St. Smooth Brick 2	D	D	D	D
5	Parking Lot Asphalt	E	E	E	E
6	Benton Indoor Carpet	F	F	F	F
7	Benton Indoor Mat Surface	G	G	F	F
8	Benton Indoor Tiles	H	H	F	F
9	Up Curb	I	I	G	G
10	Down Curb	J	I	H	G
11	High St. Rough Brick	K	J	I	H
12	Gusu Brick	L	K	J	I
13	Granite Tile	M	L	K	J
14	Asphalt Ridged Sidewalk	N	M	L	K
15	Patterned Flagstone Road	O	N	M	L
Total Number of Categories		15	14	13	12

(b) TABLE II(b): Number of instances containing features in MW and PW datasets.

Types of Data		MW	Desired PW (25% of MW)	PW
Surfaces	A	1666	417	0
	B	4524	1131	1096
	C	2085	522	853
	D	232	58	0
	E	4976	1244	1105
	F	3672	918	1293
	G	3221	806	1274
	H	3366	842	1031
	I	184	46	59
	J	185	47	56
	K	1383	346	1098
	L	554	0	0
	M	569	0	0
	N	554	0	0
	O	456	0	0

(c) TABLE II(c): The set of features for F1 and F2

	Feature	F1	F2
1	mean(acc x)	•	
2	mean(acc y)	•	
3	mean(acc z)	•	
4	var(acc x)	•	
5	var(acc y)	•	
6	var(acc z)	•	
7	mean(sum(acc))	•	•
8	mean(sum(abs(acc)))	•	•
9	var(sum(acc))	•	•
10	var(sum(abs(acc)))	•	•
11	max(sum(abs(acc)))	•	•
12	mean(gyr x)	•	
13	mean(gyr y)	•	
14	mean(gyr z)	•	
15	var(gyr x)	•	
16	var(gyr y)	•	
17	var(gyr z)	•	
18	mean(sum(gyr))	•	•
19	mean(sum(abs(gyr)))	•	•
20	var(sum(gyr))	•	•
21	var(sum(abs(gyr)))	•	•
22	max(sum(abs(gyr)))	•	•
$\sum$	Number of features	22	10

TABLE III: Subjects Used for Data Collection

Country	Parameters	Mean	Median	Mode	Min	Max
USA (MW)	Age	26.14	22.5	22.5	20	50
	Weight (Kg)	73.05	71	72	54	120
China (MW)	Age	31.4	27	47	13	47
	Weight (Kg)	54.8	55	38	38	70
USA (PW)	Age	29.71	23	23	22	55
	Weight (Kg)	65.57	66	58	58	73

TABLE IV: The Welch T-test results on group pairings

Group Pair	Surfaces		Sensor	Average p-value	Average Degree of Freedom	Average T-statistic
S3-F	F	G	Accelerometer	0.9559	3.9866	-0.024
	F	G	Gyroscope	0.4966	2.9103	0.0481
	F	H	Accelerometer	0.8819	3.9414	-0.1586
	F	H	Gyroscope	0.4910	2.7509	0.1339
	G	H	Accelerometer	0.8979	3.9387	-0.1337
	G	H	Gyroscope	0.4564	2.9055	0.0065
S2-I	J	I	Accelerometer	0.8082	3.7198	-0.2606
	J	I	Gyroscope	0.4332	3.2303	-0.8933

accurate identification provided that similar surfaces exhibit similar characteristics. The groups are shown in Table IIa.

D. Feature Extraction

A windows-based approach is used for feature extraction, which uses a window size of two seconds or 100 data points for accelerometer and gyroscope traces. Given the differences in sampling rate, the window size varies for each data set obtained from each phone. For curb data experiments, given the short duration of the experiment, the whole size of the frame is used for feature extraction. During each window size, 22 commonly used features are computed and shown in Table IIc. F1 has all computed features and is expected to give the most accurate prediction after machine learning algorithms are applied. The data collection experiment had smartphones placed in the same position on the user and wheelchair, so the mean value for x, y and z are used as features for F1. However, we argue that it is possible for crowd-sourcing contributors to not place the smartphones in the same orientation as used in baseline data collection. There is no guarantee that sensors give consistent orientation data and are physically placed exactly to all subjects in live production. Therefore, our reduced feature set, F2 does not include the features that depend on a single direction. The intention for including F2 is to see if including only multi-directional features significantly reduces the accuracy of the predictions.

The total count for instances that contain features for MW and PW experiments is shown in Table IIb. The desired number of PW features is 25% of MW features as approximately having 1000 instances for each surface category provides enough target data if transfer learning is to be performed on the more data-driven deep learning models. Overall, the number of desired PW data has been attained, but note that some surfaces were not collected due to either too low instance number or lack of access to that surface.

E. Data Resampling

A combination of oversampling using SMOTE and undersampling is performed as using a combination of the two allows for the reduction in learn time for model training as compared to solely oversampling the data, [36]. The Edited Nearest Neighbor (ENN) is used, which removes each instance of the synthetic sample for a feature class if it does not agree with K=3 neighbors [37]. Each group classification has its own different ratio of feature classes that is in the minority (lacking representation) and majority (high representation over the distribution of all the classes). In order to apply resampling, we perform a grid search for calculating the K nearest neighbors for the minority classes in each class group for the SMOTE method by analyzing accuracy, precision, recall, F1-score and ROC OVR metrics of RF and DT classifiers using a 3 time repeated stratified K-fold cross validation of 10 folds.

VI. Performance Evaluation

As described in Section IV-A, the application of machine learning models on manual wheelchair data involved the creation of eight classifiers. For source domain learning, training was performed on the set of features for F1 and F2 shown in Table IIc. The model transfer learning was performed on the models trained on S1 data and used all of the features for training as their combination resulted in higher classification accuracy.

Nonlinear algorithms, such as RF, and deep learning models often require tens of thousands of data in order to tune all of the parameters and gain enough knowledge on the particular domain to make accurate predictions. In our experiment, MW data consists of 27,627 instances of surface features, whereas PW data has 7,865 instances or about 28.47% that of MW data. Table Va shows the results for an independent T-test performed on the accuracy scores for MW-only and PW-only ANN models recorded for 10 experiments. This test is used in order to determine whether the accuracy differences are statistically significant for each group, since the models are trained on two different samples collected using different devices. The p-value for all experiments is less than 0.05, therefore accuracy scores of both models are statistically significant. We perform a comparison of the two standalone models each trained on MW and PW domain respectively with other models that use the SLT procedure.

Based on the results in Table VI, while training occurs only with minimal PW data, the PW model initially achieves the highest average surface classification accuracy at 87.79% with resampling and S3 grouping. The highest average accuracy improves to 88.41% with transfer learning methods without needing to resample the original MW data and an average of 78% or a 3.26% improvement over the minimum PW-only S1 of 74.74%. We conclude that the highest performing transfer learning model is the one that has zero fixed layers.

We performed a paired, two-sided t-test with the accuracy results from the PW-only model and the model that had zero fixed layers with transfer learning (TFL = 0). The results of the paired t-test from Table Vb indicate using unsampled data in MW training of the original model actually leads to a statistically significant improvement in transfer learning models using S1, S2 and S3 grouping. Furthermore, S3 grouping improves transfer learning accuracy on average by 10.41% using unsampled data and 8.2% using resampled data. We also note slight improvements using S2 grouping with PW-only training by 0.56%. Only when we use resampling and S3 grouping do we see a statistically significant improvement in models where no transfer learning occurred and a significant difference in the drop of performance in the PW-only model by about 2.3% and in transfer learning for 4.5%. For all three groups, the resampled models that were trained on MW-only and the transfer models all had lower accuracy than when no unsampling is applied prior to training. This could be because increasing the amount of instances really does not improve the knowledge about the original MW domain and thus leads to no improvement of performance in the transfer learning. However, it is possible that the resampling technique does aid with generalization of the knowledge represented in both PW and MW domains.

Performing the same approach to unsampled groups, we found that S1 grouping has the lowest standard deviation across the different accuracies reported as the number of layers are fixed at 0.0043. Therefore, we may conclude that even though less weights in the layers are being attuned to the new PW knowledge domain, the features are generic enough that surfaces can still be detected at a similar accuracy. The specific data from the initial source MW domain generalizes well on data from a target PW distribution. Therefore, in the cases of Unsampled S1 and Resampled S3 groups, the transfer learning models that are adapted to the PW domain share a relatively similar feature distribution as the original model adapted to the MW domain.

As the sets S2 and S3 provide statistically significant accuracies for unsampled PW-only and Transfer-TFL=0, we decided to merge them into S4 to have indoor surfaces and curbs grouped in to two surface categories (see Table VI). We achieved up to 84.69% on MW-only with an average of 83.8% and up to 89.26% on PW-only with an average of 88.1%. When we used transfer learning methods, we achieved an improvement in accuracy over training using only PW data with an average of 89.41% and up to 90.02% accuracy. This provided a statistically significant improvement as the P-Value of $3.27 * 10^{-5}$ is much less than 0.05 according to Table Vb.

Overall, the results show the feasibility of transfer learning in the domain of sensor data collection with different types of wheelchairs. Thus, previously unknown wheelchairs can be integrated into the crowdsensing environment for our ac-

(a) TABLE V(a): Independent samples T-test results for the MW-only and PW-only model accuracy scores

Group		Training Method		P-Value	Degrees of Freedom	T-Statistic
Unsampled	S1	MW only	PW only	0.000255	16.214	4.6566
	S2	MW only	PW only	$3.28*10^{-6}$	16.046	-6.9401
	S3	MW only	PW only	$4.29*10^{-9}$	16.516	-11.134
Resampled	S1	MW only	PW only	$7.05*10^{-10}$	15.716	-13.117
	S2	MW only	PW only	$4.76*10^{-9}$	16.562	-11.033
	S3	MW only	PW only	$4.99*10^{-9}$	10.76	-16.668

(b) TABLE V(b): Paired T-Test results for test accuracy scores with PW-only and highest accurate transfer learning models.

Group		Training Method		P-Value	Degrees of Freedom	T-Statistic
Unsampled	S1	Transfer TFL=0	PW only	$7.88*10^{-7}$	9	11.967
	S2	Transfer TFL=0	PW only	$7.77*10^{-5}$	9	6.8154
	S3	Transfer TFL=0	PW only	0.007954	9	3.3936
	S4	Transfer TFL=0	PW only	$3.27*10^{-5}$	9	7.6163
Resampled	S1	Transfer TFL=0	PW only	0.0741	9	2.0205
	S2	Transfer TFL=0	PW only	0.1832	9	1.442
	S3	Transfer TFL=0	PW only	$2.40*10^{-7}$	9	-13.749

TABLE VI: Mean Accuracy Across 10 Trained ANN models with Unsampled and Resampled MW Model Transfer

Performance Metric	Group		MW only	PW only	TFL=0	TFL=1	TFL=2	TFL=3	TFL=4	TFL=5
Accuracy	Unsampled	S1	0.7245	0.7474	0.78	0.7723	0.772	0.7696	0.7705	0.7676
		S2	0.7253	0.753	0.7786	0.7569	0.6919	0.676	0.5972	0.4117
		S3	0.8358	0.875	0.8841	0.8687	0.7718	0.7583	0.7474	0.5777
		S4	0.838	0.881	0.8941	0.8821	0.84	0.8266	0.7826	0.6067
	Resampled	S1	0.6953	0.7469	0.7571	0.7264	0.6568	0.6575	0.5969	0.3846
		S2	0.6996	0.7544	0.7574	0.7292	0.6601	0.6496	0.5642	0.3978
		S3	0.8128	0.8779	0.8391	0.8339	0.8431	0.8322	0.8307	0.8267

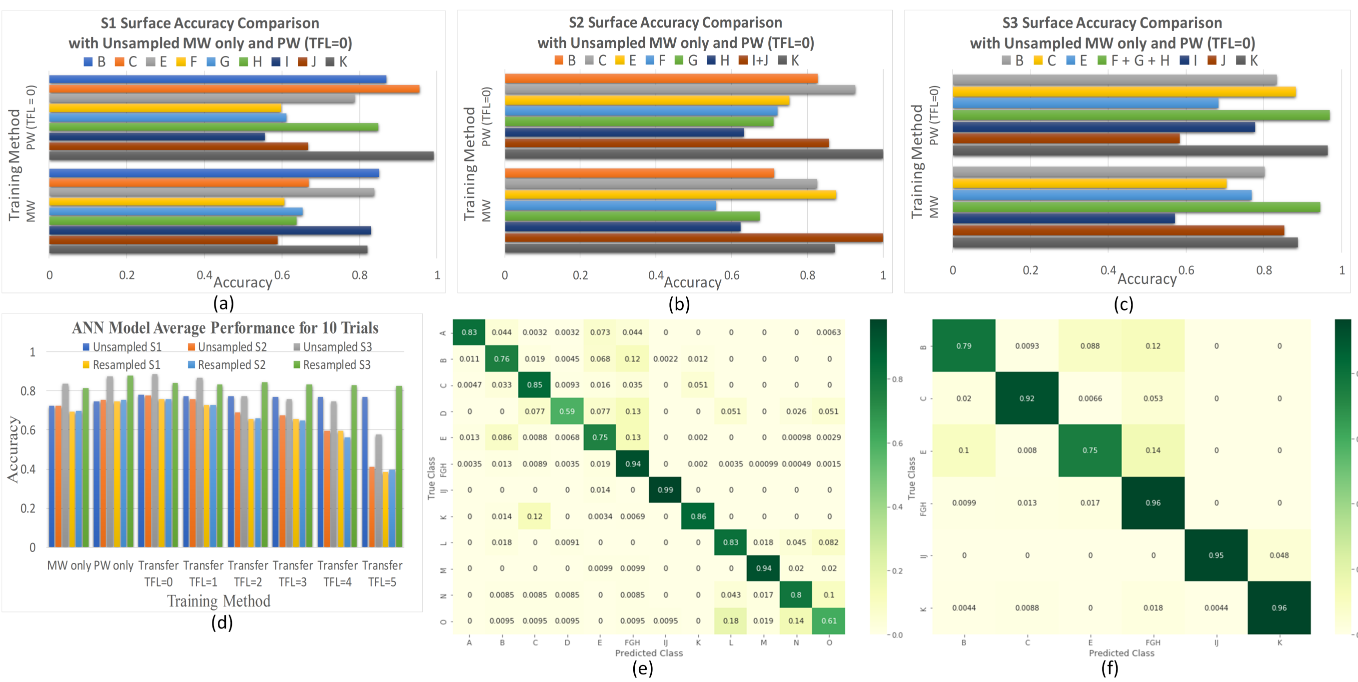

Fig. 4: Classification accuracy for each surface category (a) S1 (b) S2 (c) S3. (d) Mean Accuracy for the Modeling Strategies shown in Table VI. (e) The confusion matrix for S4 ANN MW model that is used to train the PW model confusion matrix (f).

cessible routing system. We acknowledge that some training data is required for each new wheelchair. We suggest that this data can be collected even without manual annotation. When a wheelchair collects data on previously labeled locations, this data can be considered as training data and can be used for transfer learning.

VII. Conclusion and Future Work

In this paper we investigated the problem of knowledge transfer from a machine learning based surface classification system trained using a particular type of mobility-aid device and road features to a different type of device and road features. In order to address this problem, we have adopted transfer learning that achieved successful knowledge transfer from a manual wheelchair based surface classification model to a power wheelchair. Results show that transfer learning improves accuracy of surface classification with power wheelchair data with minimal additional training data. This makes it possible to use previously unknown wheelchair types for the collection of sensor data. We have also integrated the approach into our accessible routing architecture which allows multiple users to crowdsource route information to an open-source mapping and routing engine called OpenStreetMap (OSM) while automatically identifying new surfaces using the WheelTransfer functionality. The system generates multiple accessible routes for wheelchair users between a pair of source and destination location coordinates. In the future, we want to further empower the WheelTransfer system and experiment with transfer learning on multiple other surfaces and with various other mobility aid devices. We also plan to improve the personalization features of the WheelTransfer mobile application including additional options for route selection based on user preferences and a voice enabled navigation option.

REFERENCES

[1] D. Ding, B. Parmanto, H. A. Karimi, D. Roongpiboonsopit, G. Pramana, T. Conahan, and P. Kasemsuppakorn, "Design considerations for a personalized wheelchair navigation system," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. IEEE, 2007, pp. 4790–4793.

[2] T. Völkel, R. Kühn, and G. Weber, "Mobility impaired pedestrians are not cars: Requirements for the annotation of geographical data," in *International Conference on Computers for Handicapped Persons*. Springer, 2008, pp. 1085–1092.

[3] ADA, "Americans with Disabilities Act (ADA) Standards for Accessible Design," 2018. [Online]. Available: http://www.ada.gov/2010ADAstandards_index.htm

[4] F. Zambonelli, "Pervasive urban crowdsourcing: Visions and challenges," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 578–583.

[5] N. Bicocchi, A. Cecaj, D. Fontana, M. Mamei, A. Sassi, and F. Zambonelli, "Collective awareness for human-ict collaboration in smart cities," in *22nd IEEE International WETICE Conference (WETICE 2013)*. IEEE, 2013, pp. 3–8.

[6] A. Bujari, B. Licar, and C. E. Palazzi, "Movement pattern recognition through smartphone's accelerometer," in *Consumer communications and networking conference (CCNC), 2012 IEEE*. IEEE, 2012, pp. 502–506.

[7] D. Sinkonde, L. Mselle, N. Shidende, S. Comai, and M. Matteucci, "Developing an intelligent postgis database to support accessibility tools for urban pedestrians," *Urban Science*, vol. 2, no. 3, p. 52, 2018.

[8] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1281–1293, Third 2013.

[9] V. Renaudin and C. Combettes, "Magnetic, acceleration fields and gyroscope quaternion (magyq)-based attitude estimation with smartphone sensors for indoor pedestrian navigation," *Sensors*, vol. 14, no. 12, pp. 22 864–22 890, 2014.

[10] M. O. Gani, V. Raychoudhury, J. Edinger, V. Mokrenko, Z. Cao, and C. Zhang, "Smart surface classification for accessible routing through built environment: A crowd-sourced approach," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 11–20. [Online]. Available: https://doi.org/10.1145/3360322.3360863

[11] J. Edinger, A. Hofmann, A. Wachner, C. Becker, V. Raychoudhury, and C. Krupitzer, "Wheelshare: Crowd-sensed surface classification for accessible routing," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2019, pp. 584–589.

[12] L. Beale, K. Field, D. Briggs, P. Picton, and H. Matthews, "Mapping for wheelchair users: Route navigation in urban spaces," *The Cartographic Journal*, vol. 43, no. 1, pp. 68–81, 2006.

[13] H. Matthews, L. Beale, P. Picton, and D. Briggs, "Modelling access with GIS in urban systems (magus): capturing the experiences of wheelchair users," *Area*, vol. 35, no. 1, pp. 34–45, 2003.

[14] D. Karimanzira, P. Otto, and J. Wernstedt, "Application of machine learning methods to route planning and navigation for disabled people," in *MIC'06: Proceedings of the 25th IASTED international conference on Modeling, indentification, and control*, 2006, pp. 366–371.

[15] A. D. Sobek and H. J. Miller, "U-access: a web-based system for routing pedestrians of differing abilities," *Journal of geographical systems*, vol. 8, no. 3, pp. 269–287, 2006.

[16] M. Kurihara, H. Nonaka, and T. Yoshikawa, "Use of highly accurate GPS in network-based barrier-free street map creation system," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 1169–1173.

[17] T. Völkel and G. Weber, "Routecheckr: Personalized multicriteria routing for mobility impaired pedestrians," 01 2008, pp. 185–192.

[18] C. Prandi, P. Salomoni, and S. Mirri, "mpass: Integrating people sensing and crowdsourcing to map urban accessibility," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, Jan 2014, pp. 591–595.

[19] OpenStreetMap, "Planet OSM Files," 2018. [Online]. Available: http://planet.openstreetmap.org

[20] F. Ramm, J. Topf, and S. Chilton, *OpenStreetMap: Using and enhancing the Free Map of the World*. UIT, Cambridge, 2010.

[21] D. Džafić, P. Schoonbrood, D. Franke, and S. Kowalewski, *ENav: A suitable navigation system for the disabled*. Springer, 2017, pp. 133–150.

[22] P. Neis and D. Zielstra, "Generation of a tailored routing network for disabled people based on collaboratively collected geodata," *Applied Geography*, vol. 47, pp. 70–77, 2014.

[23] M. Kawabata, R. Nishide, M. Ueda, and S. Ueshima, "Graph-based approach to context-adaptable pns and its application scenarios," *21st International Conference on Data Engineering Workshops (ICDEW'05)*, 2005.

[24] M. S. Rahaman, "Context-aware mobility analytics and trip planning," Ph.D. dissertation, RMIT University Melbourne, 2018.

[25] M. S. Rahaman, Y. Mei, M. Hamilton, and F. D. Salim, "Capra: A contour-based accessible path routing algorithm," *Information Sciences*, vol. 385, pp. 157–173, 2017.

[26] M. Sahelgozin, A. Sadeghi-Niaraki, and S. Dareshiri, "Proposing a multi-criteria path optimization method in order to provide a ubiquitous pedestrian wayfinding service," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 1, p. 639, 2015.

[27] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley *et al.*, "Unsupervised and transfer learning challenge: a deep learning approach," in *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop-Volume 27*, 2011, pp. 97–111.

[28] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Feature transfer learning for face recognition with under-represented data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5704–5713.

[29] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 193–200.

[30] T. Kamishima, M. Hamasaki, and S. Akaho, "Trbagg: A simple transfer learning method and its application to personalization in collaborative tagging," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 219–228.

[31] N. Segev, M. Harel, S. Mannor, K. Crammer, and R. El-Yaniv, "Learn on source, refine on target: A model transfer learning framework with random forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1811–1824, 2016.

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

[33] A. S. I. A. (ASIA), "American spinal injury association (asia) impairment scale," https://asia-spinalinjury.org/wp-content/uploads/2016/02/International_Stds_Diagram_Worksheet.pdf, 2019, accessed March, 2019.

[34] "Expanded disability status scale (EDSS) for multiple sclerosis," http://www.nationalmssociety.org/nationalmssociety/media/msnationalfiles/brochures/10-2-3-29-edss_form.pdf, 2020, accessed July 08, 2020.

[35] B. E. Dicianno, J. Joseph, S. Eckstein, C. K. Zigler, E. Quinby, M. R. Schmeler, R. M. Schein, J. Pearlman, and R. A. Cooper, "The Voice of the Consumer: A Survey of Veterans and Other Users of Assistive Technology," *Military Medicine*, vol. 183, no. 11-12, pp. e518–e525, 04 2018. [Online]. Available: https://doi.org/10.1093/milmed/usy033

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[37] R. Alejo, J. M. Sotoca, R. M. Valdovinos, and P. Toribio, "Edited nearest neighbor rule for improving neural networks classifications," in *International Symposium on Neural Networks*. Springer, 2010, pp. 303–310.