7-15-2019

Identifying Buildings with Ramp Entrances Using Convolutional Neural Networks

Jiawei Wu

Wenliang Hu

Joseph Coelho

Paromita Nitu

Hanna R. Paul

See next page for additional authors

Authors

Jiawei Wu, Wenliang Hu, Joseph Coelho, Paromita Nitu, Hanna R. Paul, Praveen Madiraju, Roger O. Smith, and Sheikh Iqbal Ahamed

2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (July 15-19, 2019): 74-79. DOI.

Identifying Buildings with Ramp Entrances Using Convolutional Neural Networks

Jiawei Wu
Department of Math, Stat, and Computer Science, Marquette University, Milwaukee, WI
Wenliang Hu
Department of Math, Stat, and Computer Science, Marquette University, Milwaukee, WI
Joseph Coelho
Department of Math, Stat, and Computer Science, Marquette University, Milwaukee, WI
Paromita Nitu
Department of Math, Stat, and Computer Science, Marquette University, Milwaukee, WI
Hanna R. Paul
Department of Occupational Science & Technology, University of Wisconsin-Milwaukee, Milwaukee, WI
Praveen Madiraju
Department of Math, Stat, and Computer Science, Marquette University, Milwaukee, WI
Roger O. Smith
Department of Occupational Science & Technology, University of Wisconsin-Milwaukee, Milwaukee, WI
Sheikh I. Ahamed

Department of Math, Stat, and Computer Science, Marquette University, Milwaukee, WI

Abstract:

The Americans with Disabilities Act (ADA) is a civil rights law that was signed into law in 1992 by President George H.W. Bush. The law requires wheelchair access be made available for buildings built after 1992. Buildings under the law include retail stores, hotels, banks and most other public buildings. However, there are a large percentage of buildings built before 1992 that are not wheelchair accessible. In addition, ADA does not require the location of ramp to be at the front of the building. This is an inconvenience for individuals who use wheelchairs to access a building, as a) the building may not have a ramp or b) they may have to roll around the building to where the ramp may be located. Hence, in this paper, we describe a prototype artificial intelligent system, which takes the input of a building image, and produces the output prediction for whether the building has a ramp. The system uses a deep learning technique, convolution neural network (CNN) to classify building images. We evaluated our method on a sample dataset of building images that we collected and building images from online sources. Training and validation accuracies were very high, 98.9 and 95.6 percentages respectively.

SECTION I. Introduction

To ensure equal opportunities and basic amenities for people with disabilities, it is essential to allocate fundamental resources to successfully achieve civil rights for the target group [1]. For people with a mobility disability, The Americans with Disabilities Act (ADA) requires ramp access to be made available for buildings with level changes, including retail stores, hotels, banks, and other public buildings built after 1992. Buildings built before the ADA came into effect, might not necessarily have wheelchair access to accommodate equitable entrance. In addition, the location of the ramp is not always at the front entrance of the building. The lack of a ramp can be a serious consequence for the day-to-day spontaneous movement for people with mobility disabilities. Thus, while planning a trip which involves entering a building, prior knowledge about whether a building entrance with level changes is equipped with ramps can significantly improve the experience.

The abundance of publicly available image data over recent years and the increasing importance of information retrieval, have made image processing an important source of feature extraction. Due to the development in high technology camera sensors and advanced processing capacity, a machine vision system offers efficient information advantages such as in biological image classification and social media image processing [2][3]. The manual process of classifying building entrances as having steps without ramps can be time-consuming and expensive. A machine learning algorithm for image classification can be considered as a standard tool to acquire both a cost-effective and reasonably reliable solution. Deep learning (DL) techniques provide better predictive performance and effectively deal with computational complexity and extensive run-time to acquire noise reduced image features [4]. The gradient-based training of Convolution Neural Network (CNN) is superior to traditional learning algorithms in image feature extraction.

In this project, we aim to focus on building accessibility for people with mobility impairments and propose to build an artificially intelligent system using a deep learning technique called Convolution Neural Network (CNN) to classify building images, in order to recognize whether buildings have a handicap ramp or not. If they do, those buildings are classified as accessible.

SECTION II. Background and Related Work

Building accessibility for people with disabilities is one of the requirements in ADA. Chapter 4 of the Guide to the ADA Standards specifies the guidelines and codes pertaining to the features of accessible ramps for a building, including ramp sizes and slopes [5].

However, buildings often do not have accessible ramps or lanes for people with disabilities to use as documented in major cities like New York City and Washington, D.C [6]. In one study, 554 people with disabilities surveyed revealed that twenty percent encountered difficulties in accessing a building, service or transportation at least once a day [6]. Other surveys have discovered similar findings. One study compared several convenience and grocery stores related to the number of parking spaces available for people with disabilities as well as the height of curbs and the slope of ramps [7]. McClain et al. documented restaurant wheelchair accessibility problems in another study [8]. While the literature and regulations clearly describe architectural accessibility standards, too often buildings fail to meet these standards. Moreover, no study has used machine learning method to identify the accessibility of a building [9].

Hence, this project innovatively use CNN to classify whether a building entrance has only stairs and/or has ramps as well.

The basic idea of image classification is to design a model and let it learn the differences between the different categories. Statistical based pattern recognition techniques have become mainstream especially in neural network-based image classification systems [10]. Traditional image classification algorithms are based on Bayesian techniques. However, in the 1990s Artificial Neural Network (ANN) based classifiers had become an attractive alternative. ANNs today are used for image classification as well as image processing tasks such as noise suppression and image enhancement [10]. ANNs identify features in an image based on different abstraction levels. The intensities of individual pixels are provided as input to the algorithm at the first level. At the second level, derived pixel based features are the input that determine the local features based on pixel clusters. At the third level, the relative location of local features is factored in to determine structural features like edges, corners, surfaces, etc. Structural features are combined to obtain object level features at the fourth level. At the fifth level, the order and relative location of objects provide the object set features. And finally, in the sixth level, the image can be completely described in terms of features like content, lighting, etc. [10]. Most neural-network-based image classification applications use the pixel data as features for object recognition [11]. But the limitations of ANNs include its inability to handle large number of features as well as variations in images in terms of position, orientation and scale [10].

Feed-forward networks like Convolutional Neural Networks (CNNs) in which information flow takes place in one direction only have become popular since the mid 2000s [12]. CNNs are widely used in state-of-the-art image classification applications. In several single label image classification benchmarks, their performance has surpassed human-level performance [3]. However, CNNs require a substantial amount of training data which leads to space and memory demands as well as longer training times.

Convolutional neural network was initially proposed by Yann LeCun in 1988 [9]. Different visual CNN models such as AlexNet, VGG net and GoogLeNet have been developed. Based on those nets, Jonathan Long et al. modified and fine-tuned classification nets, designing a more efficient net architecture applied to image semantic segmentation [13]. There are some teams working on enhancing the efficiency and speed of CNN computing. For example, Tian Liu et al. went deeper into the concept of CNN algorithm and they attempted to get a theoretical max speed and parallel efficiency of CNN computing by measuring and analyzing the time of forward process and back propagation computing respectively [14].

Many building and land-use classification CNN models using street and aerial view images have been developed and studied. [15] deals with classifying buildings based on their purpose (apartment, church, business, etc.) by identifying the building façade. This study focuses on building entrances.

SECTION III. Convolutional Neural Network

The neural network consists of a set of units called neurons which perform a mathematical calculation [16]. Those neurons in the different series of layers are connected to other neurons in neighboring layers.

Convolutional neural networks are made up of an input layer, multiple pairs of convolution layers and pooling layers, multiple fully-connected layers and an output layer. The layers between the input and output layers are called hidden layers. Convolution layers and pooling layers extract features from the dataset. Besides, parameters for all neurons are the same in a CNN because the same convolution filter is used to deconvolution the images and the same weights are shared [9].

A. Image RGB values

Pixels in images are usually related. For example, edges, blobs and shapes in an image or some other patterns can be signified by a certain group of pixels. Computers use 3-dimentional arrays representing images to train the CNN and then the CNN compares new images to the already trained objects. Probabilities for each category are calculated by the CNN and the category with the largest probability is treated as the predicted class.

B. Convolutional Neural Network Architecture

An example of simplified CNN architecture of this project is shown in Fig. 1.

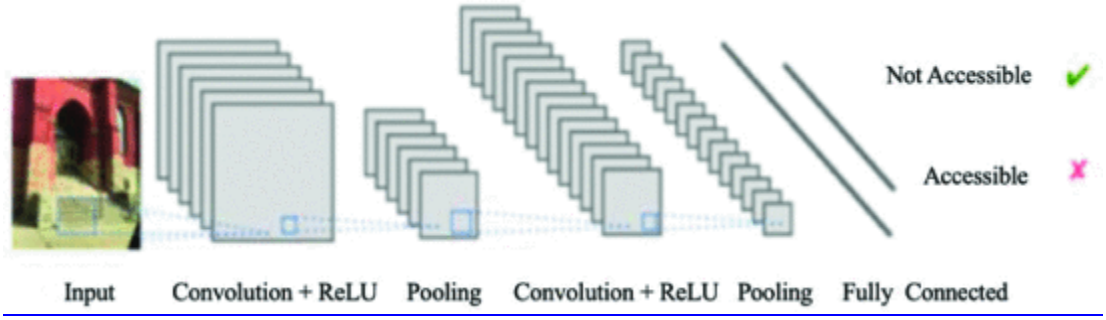

Fig. 1. Example of CNN architecture.

An image of a building's entrance is the input. Convolution layers and pooling layers are the hidden layer that scans each pixel of the image, extracts main features of the image and sends those features to the fully connected layer where the neuron will calculate the probability of each class the image belongs to according to the features. In Fig. 1, the edge of each step is the feature we need. Basically, once steps are detected in the image, the fully connected layer will give a high probability that the building entrance has steps. Typically, convolutional neural networks can comprise of several paired convolution-activation layers and pooling layers. In addition, the number of layers is modified until the model is able to produce an ideal result after training.

C. Convolution Layers

The first step in image processing is done by the convolution layer. Below shows the formula for 2D image convolution problem [17]. Layer *l* is one of convolutional layers and when it scans a 2-dimensional image with the size ot *H* x *W* and *D* channels, a 3-dimensional tensor is given as xl∈RHl×Wl×Dl which is the input of layer *l.* The output of layer *l* is x^l+1 treated as the input of layer *l* + 1. Typically, the location (row, column, channel) of each pixel is represented by i^l, j^l, d^l ($0 \le i^l \le H^l, 0 \le j^l < W^l, 0 \le d^l < D^l$)

$$x^{l+1}{}_{i^{l+1},j^{l+1},d^{l+1}} = \sum_{i=0}^{H} \sum_{j=0}^{W} \sum_{d^l=0}^{D^l} f_{i,j,d^l,d} \times x^{l}_{i^{l+1}+i,j^{l+1}+j,d^l}$$

(1)

More specifically, matrix of pixels with a certain size is multiplicated by a filter matrix (also known as kernel) with the same size, after which the multiplication values will be summed up. Then the filter slides over to the next matrix by a given number of pixels (stride) and repeats the same process until it covers all matrices of pixels [18]. Feature extraction is influenced by both the filter and the stride.

Below is an example of convolution operation with a 3 × 3 filter $\begin{pmatrix} 10-1 \\ 021 \\ -101 \end{pmatrix}$ in only one channel (Fig. 2 and Fig. 3).

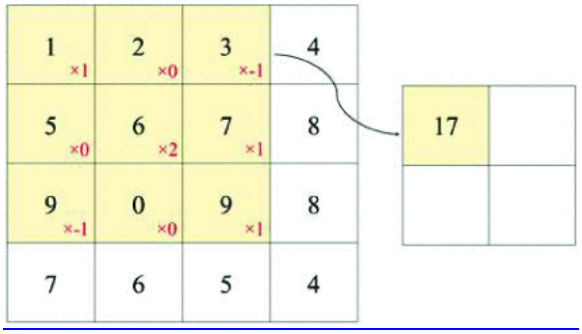

Fig. 2. (Left) Example of the first step of the image convolution operation.

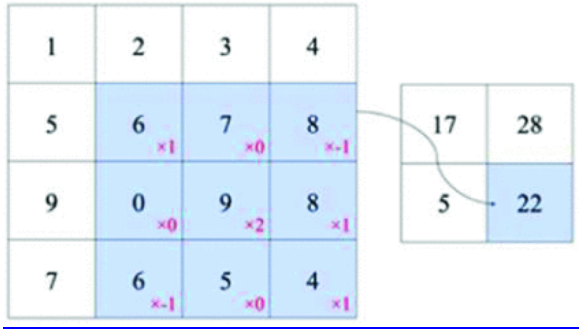

Fig. 3. (Right) Example of the last step of the image convolution operation.

D. Pooling Layer

In pooling layers, features are extracted and compressed into a small map, which simplifies the neural network computation complexity, leading to the decrease of the volume of parameters and computation[9][19]. Generally, images have a "static" property, so the main features which are useful in a certain part of an image might also be equally applicable in another part. In other words, there are only few features the model needs that are enough to represent the content in a picture approximately. Therefore, in order to describe large images, pooling is to aggregate statistics on features at different parts [20][21]. This concept is implemented in image augmentation phase as well, which will be explained later.

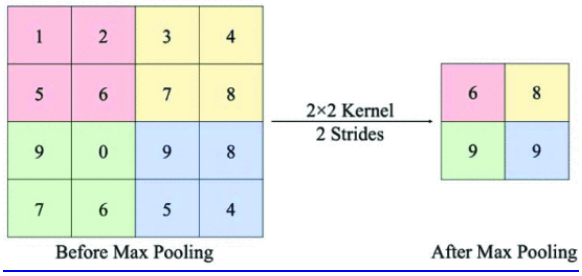

Fig. 4. Max pooling operation

In this project, max pooling is mainly used to extract features by saving the largest number in each filter and the example of max-pooling operation is shown in Fig. 4.

E. Activation Operation

Activation function is a non-linear function, also known as non-linearity mapping, that helps models to better classify features of data. Its task is to map the resulting values into the desired range. Without activation function, the model only has the capability to function as a linear map which is less complicated.

Rectified Linear Unit (ReLU) is a piecewise linear function implemented in this model. The ReLU activation function is given by:

$$ReLU = \begin{cases} x, & \text{if } n > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$$

(2)

The reason to choose ReLU as the activation function is that:

- the model of ReLU function is similar to the pattern of human brains receiving signals (Fig. 5 Left) in which the working mode of energy expenditure process is sparse and distributed [22][23].
- saturation effect will be eliminated that happens in sigmoid function.
- it changes all negative values into 0 and keeps all the positive values, which means fewer neurons are firing (sparse activation) and the network is lighter, which can enhance the efficiency of computation [9].

F. fully-connected layer

Fully-connected layer follows the last pooling layer and the output or features of the last pooling layer will be treated as input of the fully-connected layer. In this project, the fully-connected layer acts as a "classifier" for classification purposes based on features that the convolution layers generate. Pixels are aggregated and compressed as a feature map in the hidden layer and the fully-connected layer plays the role to project the feature map to a new sample data space through linear or nonlinear transformation.

G. Dropout

Dropout function is a method to reduce overfitting which refers to the model function attempting to cover all limited dataset including significant features and insignificant noise and outliers. When a model has overfitting, the accuracy of training phase will be extremely high, while it can be lower in the testing phase. Apart from reducing noise and outliers in datasets, reducing parameters in the model will reduce the probability of overfitting. The dropout function is able to reduce user-defined hyperparameter by removing neurons randomly and other neurons will represent missing neurons to do predictions. Thus, overfitting will be avoided as the model is less sensitive to specific neurons.

SECTION IV. The Application of Image Classification

The proposed model is built with CNN and has the ability to identify whether images of building entrances have only stairs or also ramps. The result indicates if the building is wheelchair friendly or not. Essentially, this is a two-class problem.

Keras is a high-level neural networks API. Once it is imported in the program, TensorFlow is used as the default backend for Keras. Sequential() in Keras is to initialize a model, into which a linear stack layers will be added. Functions Dense, Activation in keras.layer refer to setting hyperparameters for neurons in each layer and denoting an activation function. The basic procedure of building model using Keras is described below :

1. Specify the input shape: Input the size of images to the first layer and the following layers can revieve information of input size from the former layer.
2. Add layers with hyperparameters: Use function add() provided by Sequential() to add convolutional layer, pooling layer, activation function and so on.
3. Compile the model: Compilation is last step of model constraction where the optimizer and loss function are specified. Moreover, a list of metrics are available to help user evaluate the model.
4. Fire the neural network: As for fit(), it will specify the training dataset with its labels, epochs referring to the total rounds of model getting training, batch size refering to the volume of dataset model gets training with in each round.

B. Dataset

Numerous images are indispensable to train the model for image classification of accessibility of buildings. We recruited volunteers, primarily graduate students at a major university in Milwaukee to go around the town and collect pictures from buildings. Majority of the pictures focus on building's entrance were collected from Milwaukee, Chicago, Los Angeles and other urban areas by taking photos or screen-shots from Google Satellite Map. The assessment of whether there is a barrier such as a curb or stairs is the criteria of assessing the building's accessibility. Sample images of two categories are shown below (Fig. 5).

Fig. 5. Sample images of two categories. Left: accessible building. Right: inaccessible building

It is important to have a diverse set of images such as coffee shops, libraries, restaurants, university buildings, etc. to reduce overfitting and build a generic and robust system. In the original dataset, there were 803 images with different resolutions and all images, according to their category, were divided into two folders and the number of images of accessible category and inaccessible category was 407 and 369 (see Table I).

TABLE I. Numbers of Images

	Wheelchair Accessible	***Inaccessible***
Mall/Store	94	23

Private House	81	164
Apartment Complex	63	91
Restaurant/Hotel	37	6
Office	47	8
Other	85	77
Total	407	369

C. Image Pre-processing

1) Resize

In the original image dataset, the resolution of images is based on the photographing equipment. Smaller and constant size of all images are required for CNN to do image classification, because the model requires a constant input dimensionality and low resolution will speed up the model training. In the project, reasonable resolution of 150*150 pixels is applied to each image.

2) Augmentation

There are several operations of image augmentation: Zero-phase Component Analysis (ZCA), whitening (linear algebra operation to reduce the redundancy in the matrix of images), randomly shifting, randomly rotating, and horizontally or vertically flipping applied to images using Keras. Through those operations, an image can generate different types of copies. An example of image augmentation is shown by Fig. 6. Although those images including the initial one is similar, for computers, they only can identify 0 and 1, so a little change on an image will change the matrix of pixels significantly. What's more, since images have a "static" property, images look the same even if augmented, which means main features of images remain steady. Typically, the object the model needs to identify, which is the entrance of buildings, may not be centered in the images, as a result, creating augmentation training data helps the model handle off-center object images [9].

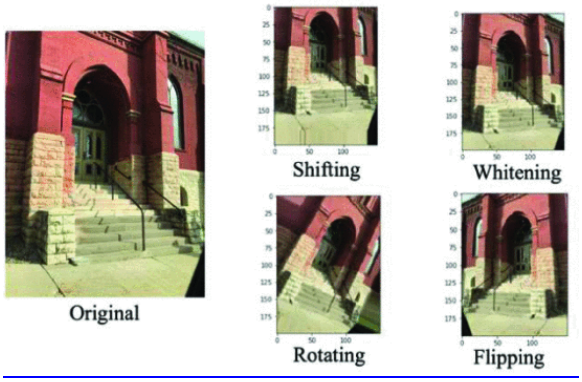

Fig. 6 Example of image augmentation

The volume of dataset increases with the help of image augmentation with Keras. Each image is randomly transformed ten times; therefore, the new dataset is composed of 8030 transformed images (4070 for accessible buildings and 3690 for inaccessible buildings). The model trained with sufficient and different types of data will be more robust and produce an accurate result.

D. Model Training

Data is split 85% for training and 15% for testing. More precisely, 6825 images are used for training the model and 1205 images are used for testing the model at each epoch when the model finishes training.

Generally, the CNN model architecture needs to be continuously debugged and hyperparameters of each operation such as convolution and pooling should be modified in the practical application until an ideal architecture with little loss and high accuracy is established.

Three CNN models are built with different numbers of hidden layers: CNN2 with 2 hidden layers, CNN3 with 3 hidden layers and CNN5 with 5 hidden layers. The architectures of these models are shown below:

In the following descriptions, hyperparameters *n* and *s* in CONV(*n*, *s*) shows there are n filters and a filter size of *s* x *s.* POOL(*z*) denotes a pooling layer with matrix size of *z.* and FC(*n*) denotes a fully-connected layer with *n* units [25].

- CNN2: CONV(64, 3), RELU, CONV(32, 3), RELU, POOL(2), CONV(64, 3), FC(128), RELU, FC(2), SOFTMAX.
- CNN3: CONV(128, 3), RELU, POOL(2), CONV(64 3), RELU, POOL(2), CONV(32, 3), RELU, POOL(2), DROPOUT(), RELU, FC(512), RELU, FC(2), SOFTMAX.
- CNN5: CONV(128, 3), RELU, CONV(128, 3), RELU, POOL(2), CONV(64, 3), DROPOUT(), RELU, CONV(64, 3), RELU, POOL(2), CONV(32, 3), DROPOUT(), RELU, POOL(2), FC(512), RELU, DROPOUT(), FC(256), RELU, FC(2), SOFTMAX

E. Result

Apart from setting different numbers of layers in the model, adjusting hyperparameters is also important for establishing a ideal model. Those hyperparameters are including the number of filters, filter's spatial extent and amount of single stride in convolution layers and pooling layers.

Each model gets training 30 epochs and the accuracy and loss of each model are shown in Table. II.

TABLE II. The Result of Model Training

	Training Accuracy	***Training Loss***	***Testing Accuracy***	***Testing Loss***
CNN2	99.66%	4.77%	79.42%	58.13%
CNN3	96.57%	10.87%	88.13%	28.85%
CNN5	98.88%	4.00%	95. 60%	13.28%

CNN5 has the best result compared with other models and the confusion matrix of testing dataset for model CNN5 is shown below in Table III. The model accuracy is 95.6%, sensitivity is 95.4%, and specificity is 95.8%.

TABLE III. Confusion Matrix

	Predictive Accessible	***Predictive Inaccessible***
Actual Accessible	600	24
Actual Inaccessible	29	552

F. Model Testing and Discussion

To test the performance and robustness of the model, 12 new images (not used in model training nor validation) were collected. Visual outputs are shown in Fig. 7. The label in the brackets is the actual value. The green display stands for correct prediction, while the red one is a prediction error. In this output, 3 predictions are wrong.

Fig. 7. Visual output of testing result

The reason why the model misclassified Fig. 8 (left) could be because the door is not front facing. The model may see the window which takes more than half of this photo as the entrance and the wall below the window as stairs. As a result, the computer produces a wrong answer, which indicates that the picture has much noise, influencing the model's judgement.

In terms of the image shown as Fig. 8 (right), the model may detect the edge between the lane and the lawn like that in inaccessible pictures. From the human point of view, the individuals with disabilities can get inside the building through the lane from a main road. However, the model after training thinks people must go in a straight line to the entrance, ignoring the possibility of the side lane to the entrance, because majority training images have the similar pattern -- a straight lane leading to buildings' door.

Fig. 8. Wrong classification of buildings' accessibility.

Similarly, in Fig. 9, the ramp may be detected by the model as a step since the ramp is built sideways. Another reason why the model mistakes the accessibility of this building is that there are not enough photos with ramps to train the model.

Fig. 9. Wrong classification of the building with a ramp.

SECTION V. Conclusion

'Accessibility' is a big domain. It includes visual, audio and/or other physical accessibility. Building accessibility by identifying stairs vs ramps is one subproject of this domain, which this project is concerned about. If people with disabilities on wheelchairs are able to enter a building without other people's help, the building is accessible for the disabled. So a curb or stairs most likely will prevent the disabled from entering the building. However, is there any possibility that wheelchairs can go over a short curb or stairs with few stepsƒ What are the criteria for stairs to count as a barrier for wheelchairsƒ Even if there is a handicap ramp that individuals with disabilities can use, the slope of the ramp should be considered, but the model lacks the ability to detect it. Apart from that, the location of entrance or ramps for people and whether the door is wide enough for wheelchairs are also important. These are all questions left for future work. In addition, the existing work can be improved by adding visual explanations for the convolutional neural network model.

In addition, collecting as many images as possible is important. It is also important to collect a diverse set of images based on different criteria such as stairs, curbs, stairs with different heights, shadows, grass, etc. We need to explore the reasons for the wrong predictions in the test set. For example, if shadows or reflections in the images are causing wrong classification, incorporating that into the model is important. Finally, presenting confidence level for each classification (probability for each class: stairs or ramp) in the test set will give a better idea of the performance of the model than accuracy and loss. The actual probability will show how close the classification is to a particular class in case of a wrong prediction. Sometimes, accuracy alone is not a good indicator for the model.

References

1."What is the Americans with Disabilities Act (ADA)ƒ | ADA National Network", [online] Available: https://adata.org/learn-about-ada.

2.G. Levi, "Age and gender classification using convolutional neural networks", [online] Available: cv-foundation.org.

3.C. Affonso, A. Rossi and F. V.-E. S. with, Deep learning for biological image classification, Elsevier.

4.A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *ImageNet Classification with Deep Convolutional Neural Networks*, 2012.

5.S. Kristensen and B. Bradtmiller, "Home - United States Access Board", *Anthropology Research Project*, [online] Available: https://www.access-board.gov/.

6."Americans With Disabilities Face Too Many Bumps in the Road - Wheelchair & Mobile Scooter Info", *Disability Resources*, 2017, [online] Available: https://www.1800wheelchair.com/news/americans-disabilities-face-many-bumps-road/.

7.L. McClain and C. Todd, "Food store accessibility", *Am. J. Occup. Ther. Off. Publ. Am. Occup. Ther. Assoc*, vol. 44, no. 6, pp. 487-491, 1990.

8.L. McClain et al., "Restaurant wheelchair accessibility", *Am. J. Occup. Ther. Off. Publ. Am. Occup. Ther. Assoc*, vol. 47, no. 7, pp. 619-623, 1993.
9.W. Hu, Image Classification using Neural Networks and Deep Learning, Marquette University, 2018.
10.M. Egmont-Petersen, D. de Ridder and H. Handels, "Image processing with neural networks—a review", *Pattern Recognit*, vol. 35, no. 10, pp. 2279-2301, Oct. 2002.
11.D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance", *Int. J. Remote Sens*, vol. 28, no. 5, pp. 823-870, Mar. 2007.
12.W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", *Neural Comput*, vol. 29, no. 9, pp. 2352-2449, Sep. 2017.
13.E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation", *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 39, no. 4, pp. 640-651, 2017.
14.T. Liu, S. Fang, Y. Zhao, P. Wang and J. Zhang, "Implementation of Training Convolutional Neural Networks", Jun. 2015.
15.J. Kang, M. Körner, Y. Wang, H. Taubenböck and X. X. Zhu, "Building instance classification using street view images", *ISPRS J. Photogramm. Remote Sens*, vol. 145, pp. 44-59, Nov. 2018.
16.B. Leibe and D. Stutz, Understanding Convolutional Neural Networks, Aachen, 2014.
17.W. Xiushen, jie xi shen du xue xi--juan ji shen jing wang luo yuan li yu shi juan shi jian (Analytical Deep Learning--Convolutional Neural Network Principles and Visual Practice), Beijing:Publishing House of Electronics Industry (PHEI), 2018.
18. Allibhai Eijaz, "Building a Convolutional Neural Network (CNN) in Keras", *Medium*, 2018, [online] Available: https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5.
19.E. S. Teaching, "CS231n Convolutional Neural Networks for Visual Recognition", 2018, [online] Available: http://vision.stanford.edu/teaching/cs231n/.
20. Abdelfattah Abdellatif, "Image Classification using Deep Neural Networks — A beginner friendly approach using TensorFlow", *Medium*, 2017, [online] Available: https://medium.com/@tifa2up/image-classification-using-deep-neural-networks-a-beginner-friendly-approach-using-tensorflow-94b0a090ccd4.
21.B. Li, C. Yang and G. Xu, "Multi-pedestrian tracking based on feature learning method with lateral inhibition", *2015 IEEE International Conference on Information and Automation ICIA 2015 - In conjunction with 2015 IEEE International Conference on Automation and Logistics*, pp. 524-529, 2015.
22.X. Glorot, A. Bordes and Y. Bengio, "Deep Sparse Rectifier Neural Networks", *JMLR W&CP*, no. 15, pp. 315-323, 2011.
23.D. Attwell and S. B. Laughlin, "An Energy Budget for Signaling in the Grey Matter of the Brain", *J. Cereb. Blood Flow Metab*, vol. 21, no. 10, pp. 1133-1145, Oct. 2001.
24. Sharma V. Avinash, "Understanding Activation Functions in Neural Networks", *Medium*, pp. 1-10, 2017.
25.S. Padmanabhan, Convolutional Neural Networks for Image Classification and Captioning, Stanford, CA, 2016.