INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

A novel approach for designing smart cane for visually impaired

Megha C. Sakkari
meghacs.15cs@saividya.ac.in
Sai Vidya Institute of Technology, Bengaluru, Karnataka

G. Krishna Chaitanya
krishnacg.15cs@saividya.ac.in
Sai Vidya Institute of Technology, Bengaluru, Karnataka

K. Raveena
raveenak.15cs@saividya.ac.in
Sai Vidya Institute of Technology, Bengaluru, Karnataka

ABSTRACT

Usually, visually impaired people use a white cane as their mobility aid along with dogs as their guide. Using technologies to make smart walking cane which will assure safe mobility of the visually impaired person and reduce the energy spent on detecting an obstacle. In this paper, the smart cane is designed which will help a visually impaired person to walk confidently. The proposed project aims at the development of an Android-based Application for smartphones in which customers can connect to a cane. The camera on the cane sends real-time images of the current surroundings and the android application works on each frame to detect upcoming obstacles. The Convolution Neural network models help in the detection of objects using Tensor Flow and stores the name of the object in a text file. Later the text in the text file is converted to voice using Google TTS (Text to Speech Conversion). Thus using advanced technologies for better living.

Keywords— *Tensorflow, Convolution neural network, Google TTS*

1. INTRODUCTION

The ability to navigate from place to place is a significant part of daily life. Human beings process the world around them mostly via the sense of sound and vision. The occurrence of the problems in the visual system can be caused by many things. Some of them are born in a state of blind, accident, illness, etc. It is a general belief that vision plays a critical role, but many would have great difficulty in identifying the visual information they use and when they use it.

We find it easy to navigate in extremely familiar places without the sense of vision. This is possible mostly due to muscle memory. This can be experienced in examples such as going to the bathroom from your bedroom in the middle of the night. But only small minorities of people have experienced navigating large-scale, unfamiliar environments without the aid of their eyes. Consider trying to catch a train in the railway station blindfolded at peak hours. Yet, the visually challenged travel independently on a daily basis. To facilitate safe and efficient navigation, blind individuals must acquire travel skills and use sources of non-visual environmental information that are rarely considered by their sighted peers. Their sense of smell and their hearing are very sharp, as they rely a lot upon these senses. They also take to feel the environment around them. This is harmless in the confines of a home. However, in an unfamiliar surroundings, this could be quite hazardous. How does one avoid running into the low-hanging branch over the sidewalk, or falling into the open manhole? When one walks down the street, how do they know when they have reached the medical store, the cafe, or their friend's house? The purpose of this chapter is to highlight some navigational technologies available to blind individuals to support independent travel.

Blind people are a term that commonly used for the people who totally blind or still have a residual vision but cannot afford their vision clearly. The occurrence of the problems in the visual system can be caused by many things. Some of them are born in a state of blind, accident, illness, etc. Blind people, usually use a cane to walk or go somewhere as a guide to know the direction and state the condition of the passing road. However, the functions of the conventional cane itself are still limited in directing and informing the obstacle to blind people especially when they are walking up to the remote destination.

The proposed system is an android application which is developed taking into account the importance of visually impaired people and energy spent on detecting the obstacles which appear in their way. The project proposes a system where an individual can navigate with ease and help in safer mobility of the visually impaired people for a better living.

The cane will have a camera placed in it which will capture real-time images. These real-time images are sent to the Android smart phone with the help of Bluetooth connectivity. The images are transferred to the android application which will detect obstacles that is image processing is done and text output is generated. The text file is converted to speech which can be heard using earphones.

The proposed project aims at the development of an Android-based Application for smart phones in which customers can connect to a cane. The camera on the cane sends real-time images of the current surroundings and the android application works on each frame to detect upcoming obstacles. The

Convolution Neural network models help in the detection of objects and stores the name of the object in a text file. Later the text in the text file is converted to voice using Google TTS (Text to Speech conversion).

2. RELATED WORK

2.1 Android platform for GUI

Android is one of the fast-growing technologies. It is being used vastly around the globe nowadays. Android powers hundreds of millions of mobile devices in more countries around the world. Android gives us a world-class platform for creating games and applications for Android users everywhere, as well as an open marketplace for distributing to them instantly. Android also gives us tools for creating apps that look great and take advantage of the hardware capabilities available on each device. It automatically adapts our UI to look its best on each device, while giving us as much control as we want over our UI on different device types. Designing the visual part and temporal behaviour of GUI is an important part of application programming in the area of human-computer interaction. It enhances the efficiency and ease of use for the underlying logical design of a program.

2.2 TensorFlow

The TensorFlow API is widely used in the field of object detection. In the era of facial recognition, it seems imperative to make use of advanced technology to recognize objects as well. TensorFlow APIs can be used to detect with bounding boxes, objects in images and or videos using either some of the pre-trained models made available or through models which you can train on your own which the API also makes it easier. TensorFlow, an open source machine library with a branch of machine learning called deep learning. It has led to proficient improvements in many zones mainly image classification and recognition.

Deep learning has a major advantage when working in the field of any texture of images. It is also done by classifiers. Classifiers are nothing but a set of codes or modules (Functions). The classifiers we use are preferably a high type of classifier namely neural network which could learn more complex functions. In order to make image classification and recognition at a comparatively easier pace, miscellaneous datasets have been created by the TensorFlow open source environment. Some illustrations include Coco, Kitti and Open image. Here comes in the coco dataset which has a collection of pre-trained detection models. They are used for initializing models when training on novel datasets and out - of - box inference. COCO (Common Objects in Context) has several features such as object segmentation, recognition in context, superpixel stuff segmentation, 1.5 million object instances, 330 K images, 80 object categories, 91 stuff categories, 5 captions per image, and 250,000 people with key points. COCO 2017 train/val browser has over 123,287 images and 886,284 instances. Deep learning is the basics of the object detection model used here. The concept of deep learning is a new area in machine learning.

We are basically training the system to learn on its own. Usually, there are three types of machine learning, supervised, semi-supervised and unsupervised learning. In this suggested system we will be using the pre-trained model which will be working on the 90 classes of COCO's dataset. Some of the models developed which use COCO are shown below.

One of the major competencies in the field of computer vision is object detection. The R-CNNSs could be classified into three advancements namely R-CNN, Fast R-CNN, and Faster R-CNN. The discrimination among these three could be done using parameters such as test time per image, speed up and map. The COCO dataset is used in Faster CNN. The Regional Convolution Neural Network plays a significant role in image segmentation and helps in the identification of the object.

Model name	Speed	COCO mAP	Outputs
ssd_mobilenet_v1_coco	fast	21	Boxes
ssd_inception_v2_coco	fast	24	Boxes
rfcn_resnet101_coco	medium	30	Boxes
faster_rcnn_resnet101_coco	medium	32	Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	slow	37	Boxes

Fig. 1: Object Detection Models

2.3 CNN model design using Tensorflow

The Basic Principle behind the working of CNN is the idea of Convolution, producing filtered Feature Maps stacked over each other. A convolutional neural network consists of several layers. Implicit explanation about each of these layers is given below.

2.4 Convolution Layer (Conv Layer)

The Conv Layer parameters consist of a set of learnable filters (kernels or feature detector). Filters are used for recognizing patterns throughout the entire input image. Convolution works by sliding the filter over the input image and along the way we take the dot product between the filter and chunks of the input image.

2.5 Pooling Layer (Sub-sampling or Down-sampling)

Pooling layer reduces the size of feature maps by using some functions to summarize sub-regions, such as taking the average or the maximum value. Pooling works by sliding a window across the input and feeding the content of the window to a pooling function. The purpose of pooling is to reduce the number of parameters in our network (hence called down-sampling) and to make learned features more robust by making it more invariant to scale and orientation changes.

2.6 ReLU Layer

ReLU stands for Rectified Linear Unit and is a non-linear operation. ReLU is an element-wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero. The purpose of ReLU is to introduce non-linearity in our ConvNet since most of the real-world data we would want our ConvNet to learn would be non-linear. Other non-linear functions such as tank *or* sigmoid can also be used instead of ReLU, but ReLU has been found to perform better in most cases.

2.7 Fully Connected layer

The Fully Connected layer is configured exactly the way its name implies: it is fully connected with the output of the previous layer. A fully connected layer takes all neurons in the previous layer (be it fully connected, pooling, or convolutional) and connects it to every single neuron it has. Adding a fully-connected layer is also a cheap way of learning non-linear combinations of these features. Most of the features learned from convolutional and pooling layers may be good, but combinations of those features might be even better.

3. DESIGN

The proposed project aims at the development of an Android-based Application for smart phones in which customers can

connect to a cane. The camera on the cane sends real-time images of the current surroundings and the android application works on each frame to detect upcoming obstacles. The Convolution Neural network models help in the detection of objects and stores the name of the object in a text file. Later the text in the text file is converted to voice using Google TTS (Text to Speech conversion).

TensorFlow is an open source software library created by Google for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets you deploy computation to one or more CPU's or GPU's in a desktop, server, or mobile device without rewriting code. TensorFlow also includes TensorBoard, a data visualization toolkit.

3.1 Image processing and CNN detection

Computers today cannot only automatically classify photos, but they can also describe the various elements in pictures and write short sentences describing each segment with proper English grammar. This is done by the Deep Learning Network (CNN), which actually learns patterns that naturally occur in photos. Imagenet is one of the biggest databases of labeled images to train the Convolutional Neural Networks using GPU-accelerated Deep Learning frameworks such as Caffe2, Chainer, Microsoft Cognitive Toolkit, MXNet, paddle paddle, Pytorch, TensorFlow, and inference optimizers such as TensorRT. Neural Networks was first used in 2009 for speech recognition and were only implemented by Google in 2012. Deep Learning, also called Neural Networks, is a subset of Machine Learning that uses a model of computing that's very much inspired by the structure of the brain. "Deep Learning is already working in Google search and in image search; it allows you to image-search a term like 'hug.' It's used to getting you Smart Replies to your Gmail. It's in speech and vision. It will soon be used in machine translation, I believe." said Geoffrey Hinton, considered the Godfather of Neural Networks. Image data preparation using Deep Learning. Preparing images for further analysis is needed to offer better local and global feature detection.

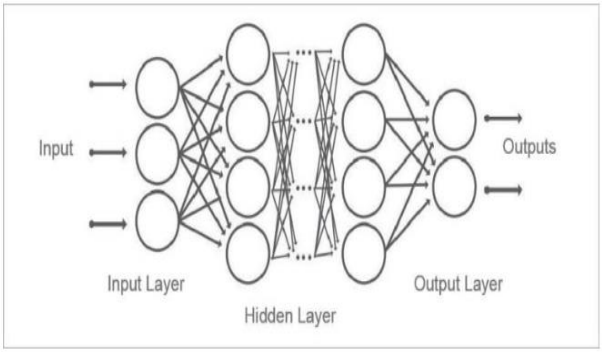

Fig. 2: Deep learning representation

3.2 Image Classification

For increased accuracy, Image classification using CNN is most effective. First and foremost, we need a set of images. In this case, we take images of beauty and pharmacy products, as our initial training data set. The most common image data input parameters are the number of images, image dimensions, number of channels, and the number of levels per pixel.

3.3 Data Labelling

It's better to manually label the input data so that the Deep Learning algorithm can eventually learn to make the predictions on its own. The objective at this point will be mainly to identify the actual object or text in a particular image, demarcating whether the word or object is oriented improperly, and identifying whether the script (if present) is in English or other languages. To automate the tagging and annotation of images, NLP pipelines can be applied. ReLU (Rectified Linear Unit) is then used for the non-linear activation functions, as they perform better and decrease training time. To increase the training dataset, we can also try data augmentation by emulating the existing images and transforming them. We can transform the available images by making them smaller, blowing them up, cropping elements etc.

3.4 Using RCNN

With the usage of region-based convolution neural network aka RCNN, locations of objects in an image can be detected with ease. Within just 3 years the R-CNN has moved from Fast RCNN, Faster RCNN to Mask RCNN, making tremendous progress towards human-level cognition of images. Below is an example of the final output of the image recognition model where it was trained by Deep Learning CNN to identify categories and products in images.

Fig. 3: Object detection example

3.5 Google TTS and Distance detection

Google Text-to-Speech is a screen reader application developed by Google for its Android operating system. It powers applications to read aloud (speak) the text on the screen which supports many languages. Text-to-Speech may be used by apps such as Google Play Books for reading books aloud, by Google Translate for reading aloud translations providing useful insight to the pronunciation of words, by Google Talkback and other spoken feedback accessibility-based applications, as well as by third-party apps. Users must install voice data for each language.

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer and can be implemented in software or hardware products. A Text-To-Speech (TTS) system converts normal language text into speech, other systems render symbolic linguistic representations like phonetic transcriptions into speech. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diaphones provide the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An

intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written words on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s.

A text-to-speech system (or "engine") is composed of two parts: a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion.
Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

4. RESULTS

We have successfully implemented a design for the smart cane which will help in guidance for visually impaired using android phone. We have used sensor flow, R-CNN, aspect ratio for distance detection and Google text to speech for implementing this paper.

5. CONCLUSION

Our model is capable of quickly identifying the objects present in the image by evaluating each region of the live feed at just a glance. Since it only requires a single pass throughout the feed, the proposed network reduces the computational cost of the evaluation, which reduces the hardware requirement, energy consumption and execution time. Using the listed advantages we can connect the camera to smart cane and computation task is done in the connected android application.

6. REFERENCES

[1] I. Endres and D. Hoiem, ‘‘Category independent object proposals,’’ in Computer Vision—ECCV. 2010, pp. 575–588.

[2] B. Alexe, T. Deselaers, and V. Ferrari, ‘‘what is an object?’’ In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2010, pp. 73–80.

[3] H. Harzallah, F. Jurie, and C. Schmid, ‘‘Combining efficient object local- ization and image classification,’’ in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep./Oct. 2009, pp. 237–244.

[4] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, ‘‘Segmentation as selective search for object recognition,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Nov. 2011, pp. 1879–1886.

[5] J. Hosang, R. Benenson, and B. Schiele. (2014). ‘‘How good are detection proposals, really?’’ [Online]. Available: https://arxiv.org/abs/1406.6962

[6] S. Maji, A. C. Berg, and J. Malik, ‘‘Classification using intersection kernel support vector machines is efficient,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2008, pp. 1–8.

[7] I. Laptev, ‘‘Improvements of object detection using boosted histograms,’’ in Proc. BMVC, vol. 3. 2006, pp. 949–958.

[8] C. L. Zitnick and P. Dollár, ‘‘Edge boxes: Locating object proposals from edges,’’ in Proc. Eur. Conf. Comput. Vis., 2014, pp. 391–405.

[9] X. Wang, M. Yang, S. Zhu, and Y. Lin, ‘‘Regionlets for generic object detection,’’ IEEE Trans. Pattern Anal. Mach. Intel. vol. 37, no. 10, pp. 2071–2084, Oct. 2015.

[10] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards real-time object detection with region proposal networks,’’ in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[11] S. Scott and S. Matwin, ‘‘Text classification using WordNet hyper- nyms,’’ in Proc. Conf. Use WordNet Natural Lang. Process. Syst., 1998, pp. 38–44.

[12] D. G. Lowe, ‘‘Object recognition from local scale-invariant features,’’ in
Proc. IEEE Int. Conf. Comput. Vis., vol. 2. Sep. 1999, pp. 1150–1157.

[13] D. Anguita, A. Boni, and S. Ridella, ‘‘Learning algorithm for nonlinear support vector machines suited for digital VLSI,’’ Electron. Lett., vol. 35, no. 16, pp. 1349–1350, Aug. 1999.

[14] S. Karnouskos and A. W. Colombo, ‘‘Architecting the next generation of service-based SCADA/DCS system of systems,’’ in Proc. 37th Annu. Conf. IEEE Ind. Electron. Soc. (IECON), Nov. 2011, pp. 359–364.

[15] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron, ‘‘A performance study of general-purpose applications on graphics processors using Cuda,’’ J. Parallel Distrib. Comput. vol. 68, no. 10, pp. 1370–1380, 2008.

[16] S. Chetlur et al. (2014). ‘‘cuDNN: Efficient primitives for deep learning.’’ [Online]. Available: https://arxiv.org/abs/1410.0759

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ‘‘ImageNet: A large-scale hierarchical image database,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2009, pp. 248–255.

[18] R. Bekkerman, M. Bilenko, and J. Langford, Eds., Scaling up Machine Learning: Parallel and Distributed Approaches. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-Based Learning Applied to Document Recognition. Piscataway, NJ, USA: IEEE Press, 2001, pp. 306–351.

[20] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, ‘‘CNN features off-the-shelf: An astounding baseline for recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2014, pp. 806–813.

[21] D. E. Williams, G. E. Hinton, and R. J. Williams, ‘‘Learning representa- tions by back-propagating errors,’’ Nature, vol. 323, no. 6088, pp. 533–538, 1986.

[22] K. Fukushima, ‘‘A neural network model for selective attention in visual pattern recognition,’’ Biol. Cybern., vol. 55, no. 1, pp. 5–15, 1986.

[23] X.-W. Chen and X. Lin ‘‘Big data deep learning: Challenges and perspec- tives,’’ IEEE Access, vol. 2, pp. 514–525, 2014.

[24] M. Abadi et al. (2016). ‘‘Tensor Flow: Large-scale machine learns- ing on heterogeneous distributed systems.’’ [Online]. Available: https:// arxiv.org/abs/1603.04467

[25] J. Bergstra et al., ‘‘Theano: A CPU and GPU math compiler in python,’’ in Proc. 9th Python Sci. Conf., 2010, pp. 1–7

