Blind Navigation System Using Artificial Intelligence

Ashwani Kumar[1], Ankush Chourasia[2]

[1,2] Dept. of Electronics and Communication Engineering, IMS Engineering College, INDIA

Abstract - *In order to provide the blind people hearable environment, this project focuses on the field of assistive devices for visual impairment people. It converts the visual data by image and video processing into an alternate rendering modality that will be appropriate for a blind user. The alternate modalities can be auditory, haptic, or a combination of both. Therefore, the use of artificial intelligence for modality conversion, from the visual modality to another.*

***Key Words*: Hearable Environment, Visual Data, Image Processing, Video Processing, Artificial Intelligence**

1. INTRODUCTION

Imagine how is the life of a blind person, their life is full of risk, they can't even walk alone through a busy street or through a park. they always need some assistance from others. They are also curious about the beauty of the world, there is excitement to explore the world, and to be aware of what is happening in front of them. Even though they can find their own things without anyone's need. The predicted cases will rise from 36 million to 115 million by 2050 for blind peoples if treatment is not improved by better funding. A growing ageing population is behind the rising numbers. Some of the highest rates of blindness and vision impairment are in South Asia and sub-Saharan Africa. The percentage of the world's population with visual impairments is actually falling, according to the study. But because the global population is growing and more people are living well into old age, researchers predict the number of people with sight problems will soar in the coming decades. Analysis of data from 188 countries suggests there are more than 200 million people with moderate to severe vision impairment. That figure is expected to rise to more than 550 million by 2050.

This project contains three main parts, a raspberry pi 3 (powered by android things), camera and artificial intelligence. When the person presses the button on device, the camera module starts to take a pictures and analyze the image using Tensorflow (open source library for numerical computation) and detect what is that picture is about, and then using a speaker or headphone, the device will voice assist the person about that picture 1).

1.1 Raspberry Pi

The Raspberry Pi was developed in the United Kingdom by Raspberry Pi Foundation. It is a series of small single board computers, basically developed to promote basic computer science in schools and other developing countries.

The Raspberry Pi can be used for various purpose. Depending on the user requirements it can be customized. We are using Raspberry Pi for image and video processing.

The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

Raspberry Pi 3 Model B has 1 GB of RAM. The Raspberry Pi 3 (wireless) is equipped with 2.4 GHz WiFi 802.11n (150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on Broadcom BCM43438 FullMAC chip with no official support for Monitor mode but implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Ethernet port.

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse. It may also be used with USB storage, USB to MIDI converters, and *virtually* any other device/component with USB capabilities. Other peripherals can be attached to the various pins and connectors on the surface of the Raspberry Pi 0.

The Raspberry Pi is connected to the system via a Rj45 cable. Raspberry pi consists of different slots for performing different functions. Raspberry pi Foundation provided a Raspbian operating system for Raspberry Pi. Python and Scratch are the main programming language used, and also support many other languages.

1.2 Raspberry Pi Camera Module

In April 2016, the original Camera Module was replaced by Raspberry Pi Camera Module. The Camera Module consists of Sony IMX219 8-megapixel sensor (compared to the 5-megapixel Omni-Vision OV5647 sensor of the original camera). Camera module can take video and still photographs. Libraries bundled in the camera can be used to create effects. It supports 1080p30, 720p60, and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi.

The camera works with all models of Raspberry Pi 1, 2, and 3. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Pi Camera Python library. The camera module is very popular in home security applications, and in wildlife camera traps.

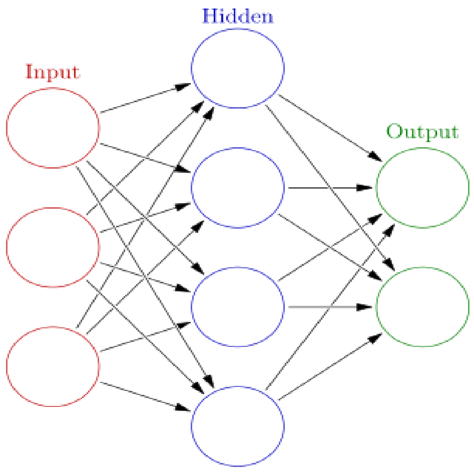
1.3 Artificial Intelligence

The field of creation of intelligent machines that work like humans and respond quickly, in computer science is known as Artificial intelligence. The core part of AI research is Knowledge engineering. Machines can react and act like humans only when they have abundant information related to the world. To implement knowledge engineering, Artificial intelligence should have access to objects, categories, properties, and relations. To initiate common sense, reasoning and problem-solving power in machines, it is a difficult and tedious task. Machine learning is another one of the core parts of AI. Learning without any kind of supervision requires an ability to identify patterns in streams of inputs, whereas learning with adequate supervision involves classification and numerical regressions. Classification determines the category an object belongs to and regression deals with obtaining a set of numerical input or output examples, thereby discovering functions enabling the generation of suitable outputs from respective inputs. Mathematical analysis of machine learning algorithms and their performance is a well-defined branch of theoretical computer science often referred to as computational learning theory.

Machine perception deals with the capability to use sensory inputs to deduce the different aspects of the world, while computer vision is the power to analyze visual inputs with a few sub-problems such as facial, object and gesture recognition.

Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks. An ANN is based on a collection of connected units or nodes called artificial neurons. Each connection (analogous to a synapse) between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it0.

In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is calculated by a non-linear function of the sum of its inputs. Artificial neurons and connections typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons have a threshold. Only if aggregate signal crosses that threshold, then the signal is sent. Artificial neurons are generally organized in layers. Different layers have different functions and perform different kinds of transformations on their inputs. Signals travel from the first (input) to the last (output) layer, possibly after traversing the layer's multiple times.



2. The model used for Image Classification: Convolutional Neural Network (CNN)

A convolutional neural network is a class of deep, feed-forward artificial neural networks that have successfully been applied to analyze the visual image.

CNNs use a multilayer perceptron's to obtain minimal preprocessing. They are also known as space invariant artificial neural networks (SIANN), due to their shared-weight architecture and translation invariance characteristics. The deep convolutional neural network can achieve reasonable performance on hard visual recognition tasks, matching or exceeding human performance in some domains. This network that we build is a very small network that can run on a CPU and on GPU as well.0.

CNN is composed of convolutional modules of the stack that performs feature extraction. Each module has a convolutional layer followed by a pooling layer. The last convolutional module is followed by single or denser layers that perform classification. The final dense layer in CNN contains a single node for each target class in the model, with a softmax activation function to generate a value between 0–1 for each node (the sum of all these softmax values is equal to 1). We can interpret the softmax values for a given image as relative measurements of how likely it is that the image falls into each target class.

Convolutional layers, which apply a specified number of convolution filters to the image. For each sub-region, mathematical operations are performed by layer to produce a single value in the output feature map. Convolutional layers then typically apply a ReLU activation function to the output to introduce nonlinearities into the model.

The rectifier is an activation function defined as the positive part of its argument:

f(z)= z+ = max(0,z), where z is the input to a neuron.

A smooth approximation to the rectifier is the analytic function f(z) = log (1+ exp z), which is called the softplus function. The derivative of softplus is

f'(z)= exp (z) / (1+exp z) = 1/(1+exp(-z))

i.e. the logistic function.

Convolutional Layer: Applies 32 5x5 filters (extracting 5x5-pixel subregions), with ReLU activation function.

We are using CIFAR-10 classification to classify RGB 32x32 pixel images. The reason CIFAR-10 was selected was that it is complex enough to exercise much of TensorFlow's ability to scale to large models. At the same time, the model is small enough to train fast, which is ideal for trying out new ideas and experimenting with new techniques.

Model Architecture

The model CIFAR-10 is a multi-layer architecture consisting of alternating convolutions and nonlinearities. These layers are followed by fully connected layers leading into a softmax classifier 0. This model achieves a peak performance of about 86% accuracy within a few hours of training time on a GPU. It consists of 1,068,298 learnable parameters and requires about 19.5M multiply-add operations to compute inference on a single image. By using CIFAR-10 Model the images are processed as follows:

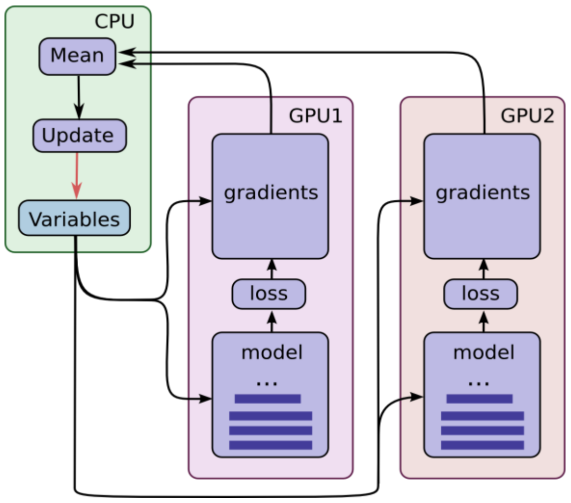
1. They are cropped to 32 x 32 pixels, centrally for evaluation or randomly for training.

2. They are approximately whitened to make the model insensitive to dynamic range.

This is a good practice to verify that inputs are built correctly.

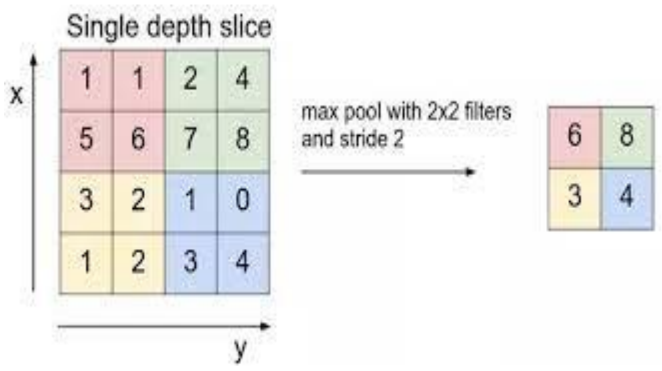
Reading images from disk and distorting them can use a non-trivial amount of processing time. To prevent these operations from slowing down training, we run them inside 16 separate threads which continuously fill a Tensor Flow queue.

Here is a diagram of this model:



Pooling layers, which down sample the image data extracted by the convolutional layers to reduce the dimensionality of the feature map in order to decrease processing time. We used max pooling algorithm, which extracts sub-regions of the feature map (e.g., 2x2-pixel tiles), keeps their maximum value, and discards all other values 0.

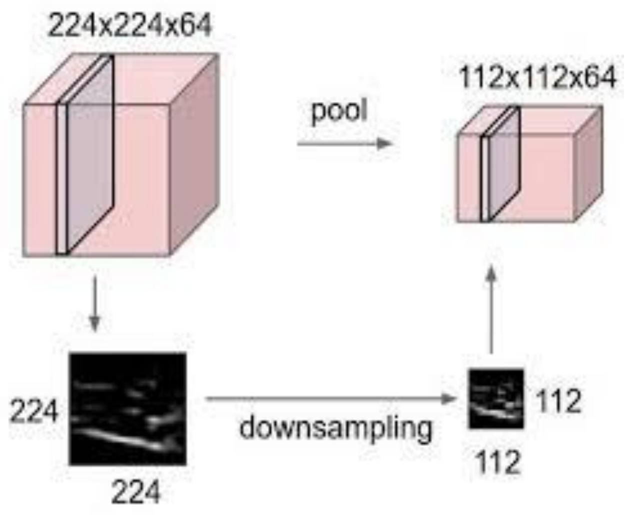
The most common form of pooling is Max pooling where we take a filter of size F*F and apply the maximum operation over the F*F sized part of the image.



If you take the average in place of taking maximum, it will be called average pooling, but it's not very popular. If your input is of size w1*h1*d1 and the size of the filter is f*f with stride S. Then the output sizes w2*h2*d2 will be:

W2= (w1-f)/S +1
h2=(h1-f)/s+1
d2=d1

Most common pooling is done with the filter of size 2*2 with a stride of 2. As you can calculate using the above formula, it essentially reduces the size of input by half 0.



Dense (fully connected) layers, which perform classification on the features extracted by the convolutional layers and down sampled by the pooling layers. In a dense layer, every node in the layer is connected to every node in the preceding layer.

Dense Layer 1: 1,024 neurons, with dropout regularization rate of 0.4 (probability of 0.4 that any given element will be dropped during training)

Dense Layer (Logits Layer) 2: 10 neurons, one for each digit target class (0–9).

Logits Layer, the final layer of our neural network is the logits layer, which will return the raw values for our predictions. The logit model is a regression model where the dependent variable (DV) is categorical. where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail or win/loss. Cases, where the dependent variable has more than two outcome categories, may be analyzed in multinomial logistic regression, or, if the multiple categories are ordered, in ordinal logistic regression 0.

Our final output tensor of the CNN, logits, has shape [*batch size, 10*]

Generate Predictions, the logits layer of our model returns our predictions as raw values in a [*batch size, 10*]-dimensional tensors. Let's convert these raw values into two different formats that our model function can return:

The predicted class for each example: a digit from 0–9.
The probabilities for each possible target class for each example: the probability that the example is a 0, is a 1, is a 2, etc.

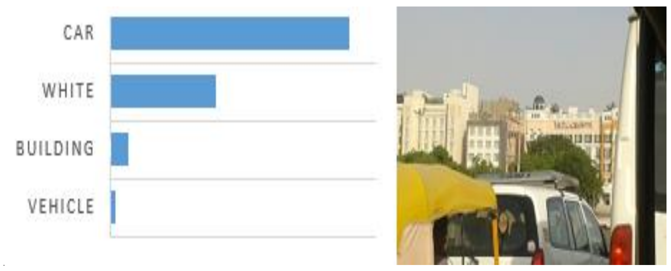
Calculate Loss, for both training and evaluation, we need to define a loss function that measures how closely the model's predictions match the target classes. To calculate cross entropy, we use One hot encoding technique. One-hot is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0). This cost that will be minimized to reach the optimum value of weights.

Configure the Training Op, we configure our model to optimize this loss value during training. We'll use a learning rate of 0.001 and stochastic gradient descent as the optimization algorithm 0.

This is a small network and is not state-of-the-art to build an image classifier but it's very good for learning especially when you are just getting started. For our training set, we get more than 90% accuracy on the validation set. As we save the model during training, we shall use this to run on our own images.

3. CONCLUSIONS

The goal of this research is to provide the better image processing using artificial intelligence. By using CNN image classifier, we predict the correct answer with more than 90% accuracy rate. By doing so we achieved the state-of-art result on the CIFAR-10 dataset. We also use the trained model with real time image and obtained the correct label.



We integrated tensorflow in Android studio with our trained model. And it is deployed in raspberry pi with the help of Android Things Operating System.

REFERENCES

1) Dean, Jeff; Monga, Rajat; et al. (9 November 2015). "TensorFlow: Large-scale machine learning on heterogeneous systems" (PDF). TensorFlow.org. Google Research. Retrieved 10 November 2015.

2) Getting Started with Raspberry Pi; Matt Richardson and Shawn Wallace; 176 pages; 2013; ISBN 978-1449344214.

3) Raspberry Pi User Guide; Eben Upton and Gareth Halfacree; 312 pages; 2014; ISBN 9781118921661.

4) Xilinx. "HDL Synthesis for FPGAs Design Guide". section 3.13: "Encoding State Machines". Appendix A: "Accelerate FPGA Macros with One-Hot Approach". 1995.

5) Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; Kingsbury, B. (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition --- The shared views of four research groups". IEEE Signal Processing Magazine.**29**(6):82-97. doi:10.1109/msp.2012.2205597

6) R. J. Williams and D. Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. In Back-propagation: Theory, Architectures, and Applications. Hillsdale, NJ: Erlbaum, 1994

7) Hilbe, Joseph M. (2009), Logistic Regression Models, CRC Press, p. 3, ISBN 9781420075779.

8) Zoph, Barret; Le, Quoc V. (2016-11-04). "Neural Architecture Search with Reinforcement Learning". arXiv:1611.01578

9) Hope, Tom; Resheff, Yehezkel S.; Lieder, Itay (2017-08-09). Learning TensorFlow: A Guide to Building Deep Learning Systems. "O'Reilly Media, Inc.". pp. 64–. ISBN 9781491978481.

10) Sutton, R. S., and Barto A. G. Reinforcement Learning: An Introduction. The MIT Press, Cambridge, MA, 1998

11) Zeiler, Matthew D.; Fergus, Rob (2013-01-15). "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks". arXiv:1301.3557

12) Lawrence, Steve; C. Lee Giles; Ah Chung Tsoi; Andrew D. Back (1997). "Face Recognition: A Convolutional Neural Network Approach". Neural Networks, IEEE Transactions on. 8 (1): 98–113. CiteSeerX 10.1.1.92.5813

13) Krizhevsky, Alex. "ImageNet Classification with Deep Convolutional Neural Networks". Retrieved 17 November 2013.

14) Yosinski, Jason; Clune, Jeff; Nguyen, Anh; Fuchs, Thomas; Lipson, Hod (2015-06-22). "Understanding Neural Networks Through Deep Visualization". arXiv:1506.06579

15) Graupe, Daniel (2013). Principles of Artifircial Neural Networks. World Scientific. pp. 1–. ISBN 978-981-4522-74-8.

16) Dominik Scherer, Andreas C. Müller, and Sven Behnke: "Evaluation of Pooling Operatiorns in Convolutional Architectures for Object Recognition," In 20th International Conference Artificial Neural Networks (ICANN), pp. 92-101, 2010. https://doi.org/10.1007/978-3-642-15825-4_10

17) Graupe, Daniel (7 July 2016). Deep Learrning Neural Networks: Design and Case Studies. World Scientific Publishing Co Inc. pp. 57–110. ISBN 978-9r81-314-647-1

18) Clevert, Djork-Arné; Unterthiner, Thomas; Hochreiter, Sepp (2015). "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". arXiv:1511.07289

19) Yoshua Bengio (2009). Learning Deep Architectures for AI. Now Publishers Inc. pp. 1–3. ISBN 978-1-60198-294-0.

20) Christopher Bishop (1995). Neural Networks for Pattern Recognition, Oxford University Press. ISBN 0-19-853864-2

BIOGRAPHIES

Ashwani Kumar, born in 1996, India. He is currently pursuing B.tech in ECE from IMS Engineering College. His interest is in Data Science and Machine Learning.

Ankush Chourasia, born in 1996, India. He is currently pursuing B.tech in ECE from IMS Engineering College. His interest is in developing Android Application.