

LOCATION NODE IDENTIFICATION IN INDOOR NAVIGATION

INTRODUCTION

The aim of this work is to detect and classify the location inside for indoor navigation in a image. Deep learning model is developed to classify the location node from a given set of location nodes in indoor space. We collected more than 3000 images currently inside our home to train DCNN model and acheived accuracy of 92% on validation set and accuracy of 78% on test set.

DATASET

For collecting dataset we created a video at location node by shooting space all around, covering all angels of view. We converted video format to [480p@30fps](#). Then extracted images from all frames of video and rescale them to 256x256. After that remove low quality images with blur less than 0.35. We randomly split the dataset into two: 150 images from each category as validation set and use the rest as training dataset.

EXPERIMENT

We trained our dataset based on Caffe framework. We finetuned our dataset on AlexNet pretrained model. It was trained using stochastic gradient descent with a base learning rate of 6-2, reduced by 50% every 2000 iterations and with momentum of 0.9. The model is deployed on Amazon Web Service with a g2.2xlarge type instance. The training process takes 2 hours.

On the test dataset our model achieves 78% accuracy.

APPROACH

We hosted our model using Python Flask framework on apache web server on AWS. Everytime during navigation we call the hosted model from our Android App by sending the image to it then It returns the indoor Location. Based on the indoor location returned We can find out the node in indoor space we have reached.

FUTURE WORK

Currently due to large size of our model we need to host it over the cloud server. Due to which our navigation don't work perfectly when moving at higher speed. So we are trying to train SqueezeNet which can make our model less than 4mb in size or some other mobile compatible models . By this we can use our small size model directly on Android phone hardware instead of hosting it on cloud and get more speed of navigation.

