Audio Description and Content Moderation App

Summer Undergraduate Research Award

Ayush Patel
2016CS10396
Computer Science
CGPA: 9.305
Mob: 9891052662
cs1160396@iitd.ac.in

Mohit Gupta
2016CS50433
Computer Science
CGPA: 9.579
Mob: 9466479674
cs5160433@iitd.ac.in

Supervisor:-
Aaditeshwar Seth
Professor
Department of CSE
aseth@cse.iitd.ac.in
IIT Delhi

Prof. S. Arun Kumar
Head of Department
Department of CSE
sak@cse.iitd.ernet.in

1 Introduction

Citizen sourced journalism is an important medium of expression for people, especially in rural India, who are often kept out of discussions regarding issues faced by them. With increasingly improved technology and the penetration of mobile phones, giving people lot of computation power in their hands, the medium of the contribution of stories by the users is no longer limited to just text-based. People can also now contribute audios, videos, and photos to support their stories. But the underlying fact remains that whatever the degree of participation, and whatever the medium, there is a strong need of screening the contributed items, identifying the topics of discussion and grouping based on the same before they can be published. This is necessary to maintain the quality of the news of being reported, the decisions being carried out and take into account the sensitivities of various parties.

Our app is built around the Mobile-Based community platforms like ———— which use the common "missed call" concept where users place a call to a phone number and the server cuts the call and calls them back, thus making the system free of cost for the users. The IVR presents options to record voice messages ———— they want to share. A wide variety of topics can be featured on this app, including hyperlocal news, job openings, agriculture advisory, social issues such as early marriage and domestic violence, health QA, governance and accountability, folk songs and poems, and local and national level advertisements. The bulk of the content on the platform is user-generated with recordings contributed by the IVR itself and subsequently moderated and curated by the content team for publication on the IVR.

As the number of stories contributed by people rises, the need for moderation can turn into a bottleneck for scaling such platforms. It is important to ensure that the items worth publishing are accessible by other people soon after they are recorded. Our app aims to solve this problem by easing the task of the content moderation. In this scheme, moderation is done at two levels: first by community representatives with the help of the smart-phone app and second, by a central team of dedicated moderators. Different responsibilities are split across the two levels, as explained later.

According to surveys, in most of these current platforms, for more than 70% of the rejected messages, the main reason for rejection is poor audio quality. Other messages get rejected because the report is not articulate enough, or it is incomplete, and only 1.5% messages are rejected because the content is objectionable or incorrect. Our app also aims at addressing these problems. The MV content moderators are presented with this app which takes in all the recorded audios and categorizes each audio based on its quality by comparing it with a minimum threshold. This app then tries to enhance the poor-quality audios, and the ones who can't be processed further are discarded. The app also has an abusive filter which follows advanced editorial policies to ensure that abusive words and statements are removed from content.

Currently, all that the community reporter sees is a long list of audios coming from the IVR. We also aim to add certain features to our app by which the stories contributed by the community reporter are better organized and hence aid him to follow-up on the issue much more effectively. Data and information in such an organized form can then also be shown to various concerned authorities, hence making the voices of the people to be heard where they need to be and improve chances of a resulting impact.

The app associates tags based on the broad topic/theme the audio belong to and thus organizes the data based on these tags. These tags help grouping similar kind of data and thus, a particular moderator specialized in that topic can go for the second level of screening of the same.

2 Objectives

Design, build and validate an app to moderate and curate the voice messages recorded on IVR (Interactive Voice Response) systems incorporated in Mobile-Based Community Platforms.

The main objectives of the app is to satisfy the need of automatic content moderation in various Community Platforms by adding features to enhance audio quality, convert to text, abusive content filters, associates tags based on the broad topic/theme(and subtopics having to rank) the audio belongs to and thus, to organize audios collected via the app into issues and enable better tracking of the status of each.

3 Approach to the project

The basic approach is to train our model using audios that already have been segregated into different categories, so that our networks "learns" to segregate audios.
In order to achieve this we have broken down our problem in several parts. First we will be giving tags to the audios depending on their quality and will try to enhance the quality of poor quality audio. We will reject the audios for which no enhancement is possible. Then the audio is converted to text using Natural language processing (NLP). Using the textual information, we provide hash-tags to the audio and then using a multiclass neural network we assign the probability of the audio falling under a particular topic and then assigning it the main topic. Then depending on the topic, we decide the abusive tolerance level and then mark the audios with abusive level more than the tolerance level for consideration from moderator. Depending on the scores of the moderator, if the combined score of the moderators who mark the audio as "Rejected" is above a certain level, then the audio is moved to trash. The scores to the moderators (and community reporters) is given using a standard scoring algorithm which will be developed in future. Then after passing through all the stages, the audio is available to the user through the app.

1. Audio Quality
    (a) Set up a deep neural network with pre-trained weights to separate noise from the audio.
    (b) Depending on the output audio file, if the audio is not clear than the audio file is rejected.
2. Audio Enhancement
    (a) We will use digital sampling to reconstruct the audio as much as possible.
    (b) We sample at least twice as fast as the highest frequency we want to record so that we can use Nyquist theorem perfectly reconstruct the original sound wave from the spaced-out samples.
3. Speech Recognition and abuse filtering
    (a) The user is asked to choose one of the provided languages so that it helps us to use a pre-trained neural network to convert audio to text.
    (b) We will be using APIs like Watson speech to text or Google speech aip for converting the audio to text.
    (c) Depending on the abusive words in the language, we segregate the audios into abusive tolerable or abusive intolerable.
4. Topic Assignment
    (a) We use "one vs all" multi-class classifier neural network for the purpose of hash-tag assignment and topic assignment to the audio.
    (b) The moderators (or community reporters) may change the topic is the want to do so.
5. Further Possibilities
    (a) We will be training on more languages to accommodate linguistic diversity prominently in rural areas
    (b) Apart from the app for content moderators, we may also build an app for users so that they can get news on demand as per their preferences.

4 Budget and duration

4.1 Budget

No budget is required for this project.

4.2 Duration

We will try to complete this project by the end of the summer break i.e. the end of July, 2017.

5 Background

5.1 Deep Learning

Deep Learning is a branch of machine learning in which multiple parameter based models are used in series. In a deep network, there are many layers between the input and output, allowing the algorithm to be executed in multiple processing steps, composed of **multiple linear and non-linear transformations**. At each layer, the signal is transformed by a processing unit, like an artificial neuron, whose parameters are **'learned'** through training. Deep Learning has been shown to excel in tasks where the goal is to find **intuitive** patterns in the data.[**?**] In particular, in the field of Computer Vision, deep networks are increasingly used to extract **feature descriptions and inter-relationships between features** from images.[**?**]

5.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN, or ConvNet) are a type of feed-forward artificial neural network in which the connectivity pattern between the neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the **receptive field**. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a **convolution operation**. A Convolutional Neural Network consists of the following layers.[**?**]

5.2.1 Convolutional Layer

The convolution layer is the core building block of a CNN. The layer's parameters consist of a set of **learnable filters** (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter.[**?**] As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

5.3 Training Deep Neural Networks

A Deep Neural Network is at it's core a parameter based function. All of these parameters are **trained** automatically from inputs and expected output tuples (training data). The

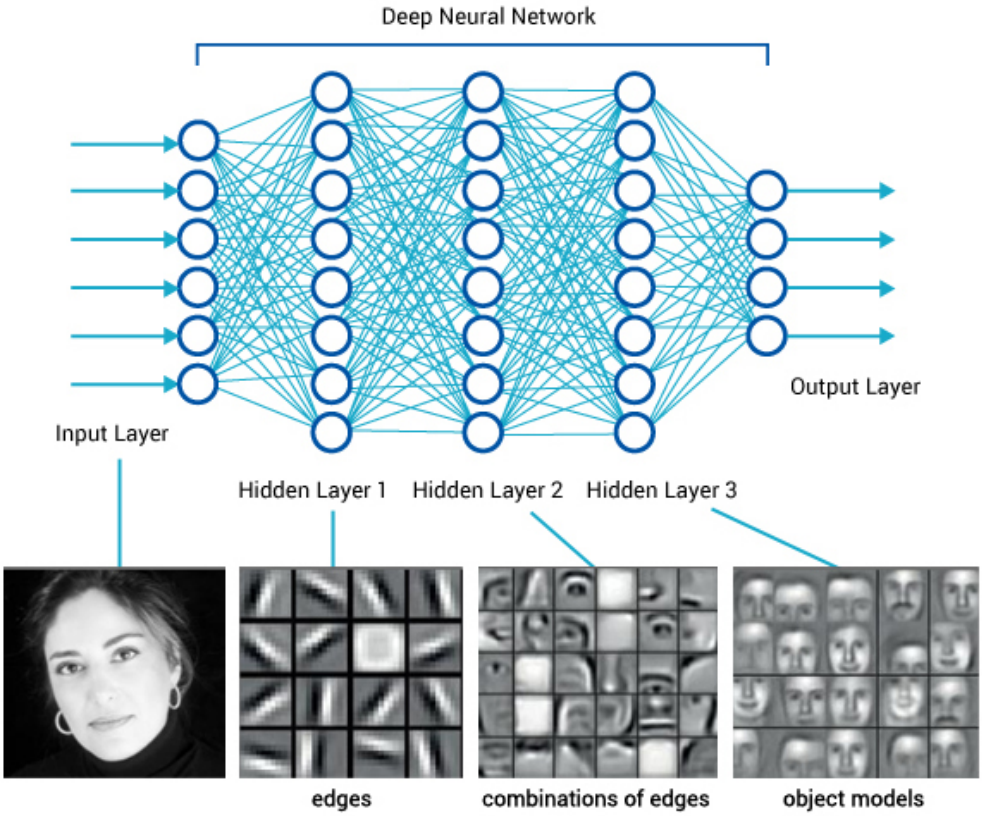

Figure 1: Illustration of Deep Learning as applied to Vision

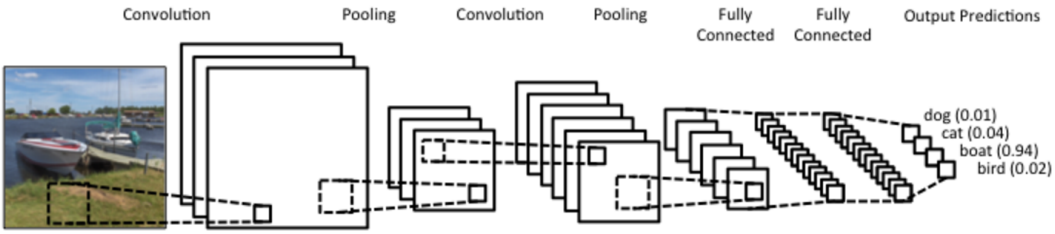

Figure 2: A Typical Convolutional Neural Network

training process revolves around minimizing a particular cost function using methods like **Stochastic gradient descent**. The input is given to the network in a feed forward fashion and the parameters are modified from the last layer to the first **(Backpropagation)**. Neural Networks, by design, require huge amounts of training data and take a large time

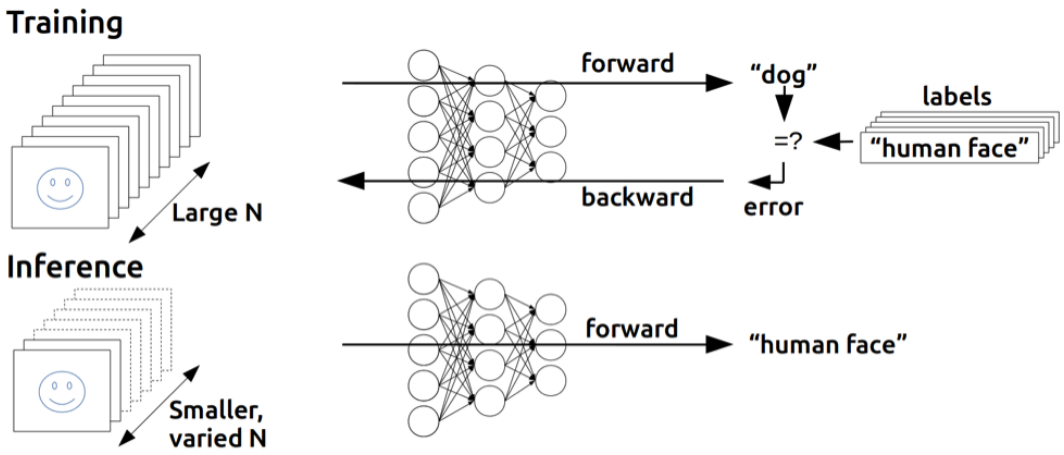

Figure 3: Training and Inference Processes

to get trained. For some perspective, most current state of the art image classifiers have $>$ 100 million parameters and are trained on more than 1.2 million images.