Punyaslok Sarkar, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 10, Issue 5, (Series-IV) May 2020, pp. 63-77

RESEARCH ARTICLE OPEN ACCESS

Object Recognition with Text and Vocal Representation

Punyaslok Sarkar, Mrs. Anjali Gupta
4th year student, Computer Science & Engineering CMR Institute of Technology
Asst. Professor,Computer Science & Engineering CMR Institute of Technology

ABSTRACT
The YOLO design enables end-to-end training and real-time speeds while maintaining high average precision. In present industry, communication is the key element to progress. Passing on information, to the right person, and in the right manner is very important, not just on a corporate level, but also on a personal level. The world is moving towards digitization, so are the means of communication. Phone calls, emails, text messages etc. have become an integral part of message conveyance in this tech-savvy world. In order to serve the purpose of effective communication between two parties without hindrances, many applications have come to picture, which acts as a mediator and help in effectively carrying messages in form of text, or speech signals over miles of networks. Most of these applications find the use of functions such as articulatory and acoustic-based speech recognition, conversion from speech signals to text, and from text to synthetic speech signals, language translation among various others. In this review paper, we'll be observing different techniques and algorithms that are applied to achieve the mentioned functionalities. Communication is the main channel between people to communicate with each other.

Date of Submission: 18-05-2020 Date of Acceptance: 03-06-2020

I. INTRODUCTION

Over the past few years, Cell Phones have become an indispensable source of communication for the modern society. We can make calls and text messages from a source to a destination easily. It is known that verbal communication is the most appropriate modem of passing on and conceiving the correct information, avoiding misquotations. To fulfil the gap over a long distance, verbal communication can take place easily on phone calls. A path-breaking innovation has recently come to play in the SMS technology using the speech recognition technology, where voice messages are being converted to text messages. Quite a few applications used to assist the disabled make use of TTS, and translation. They can also be used for other applications, taking an example: Siri an intelligent automated assistant implemented on an electronic device, to facilitate user interaction with a device, and to help the user more effectively engage with local and/or remote services [1] makes use of Nuance Communications voice recognition and text-to-speech (TTS) technology. In this paper, we will take a look at the different types of speech, speech recognition, speech to text conversion, text to speech conversion and speech translation. Under speech the recognition we will follow the method i.e. pre-emphasis of signals, feature extraction and recognition of the signals which help us in training and testing mechanism. There are various models used for this purpose but Dynamic time warp, which is used for feature extraction and distance measurement between features of signals and Hidden Markov Model which is a stochastic model and issued to connect various states of transition with each other is majorly used. Similarly for conversion of speech to text we use DTW and HMM models, along with various Neural Network models since they work well with phoneme classification, isolated word recognition, and speaker adaptation. End to end ASR is also being tested as of late 2014 to achieve similar results. Speech synthesis works well in helping convert tokenized words to artificial human speech.

Relevance of the Project

It is widely used in computer vision taskssuch as face detection, face recognition, video objectco-segmentation. It is also used in tracking objects, for example tracking a ball during a footballmatch, tracking movementof a cricket bat, or tracking a person in a video

Every objectclass has its own special featuresthat helps in classifying the class – for example all circlesare round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e. the centre) are sought. Similarly, when looking for squares, objects that

are perpendicularat corners and have equal side lengths are needed. A similar approach is used for faceidentification where eyes, nose, and lips can be found and featureslike skin colour and distance between eyes can be found.

1.1 Scope of the Project

In this project we are using general purpose and, a unified model for object detection object (YOLO). The object is detected using the Yolo algorithm and objects name is converted from text to speech

1.2 Chapter Wise Summary

Normally, a blind person uses cane as a guide of him to protect him from obstacles. Most of area of surrounding is covered by the cane, especially the area near to his legs like stairs etc. But certain areas such as near to his head, especially when he is entering or leaving the door which is short in height. This system is specially designed to protect the area near to his head. The product is designed to provide full navigation to user into the environment. It guides the user about obstacles as well as also provides information about appropriate or obstacle free path. We are using buzzer and vibrator, two output modes to user. Logical structure: The logical structure of our system is shown in following fig 1. The can be divided into three main parts: the user control, sensor control, and the output to the user. Fig 1. Logical Structure The user control includes the switches that allow the user to choose project's mode of operation. There are basically two modes of operation, Buzzer mode and Vibration mode. These modes are provided to user for taking output on his portability. Sometimes, he is not comfortable in getting the output in one mode. Vibration mode always not comfortable, can irritate him. Similarly, when there is a lot of noise in environment the buzzer mode is not portable. Another switch is controlled by the user, called initializing switch. The initializing switch is pressed when the user wants to stop the system. Sensor control determines when to tell the sensor to take a measurement and receives the output from the sensor and normalizes it to control value for the sensors. Basically, we are designing a sensor module. We are using proximity IRsensor for detection and it is mounted on a stepper motor. Stepper motor rotates continuously with an angle of 90 degree. The 90 degree angle is divided into three 30 degree portions. Two 30 degree areas are for indicating left direction or right direction obstacles, and third 30 degree area is for indication front obstacles. The main thing is our system is based on protecting the near head area because walking cane does not protect this area. Output to the user includes the indication of obstacles to user. Basically we are using two output modes, vibration mode and buzzer mode. User can select any of the two modes in accordance to his convenience. Sometimes vibration mode is portable for him, especially when there is a lot of noise into the environment. Buzzer mode is generally used when the environmental noise is low and sometimes vibration can create irritation to the user. Architecture: The system architecture diagram of our project is given in following Fig 3. There are certain functions accomplished by these blocks. The description of blocks is as following Fig 3.Block Diagram As per our propose application blind person taking video of the path where he was walking the application will give voice message to that blind person and it will help to that person for identifying he's path . The object gets detected by the key matching technique which is used in the algorithm. And match that object with the database images to confirm the obstacle that comes into the way. When object is matched with database objects the application gives the voice instruction by using the Speech synthesizer. So, Blind user gets the direction from the application.

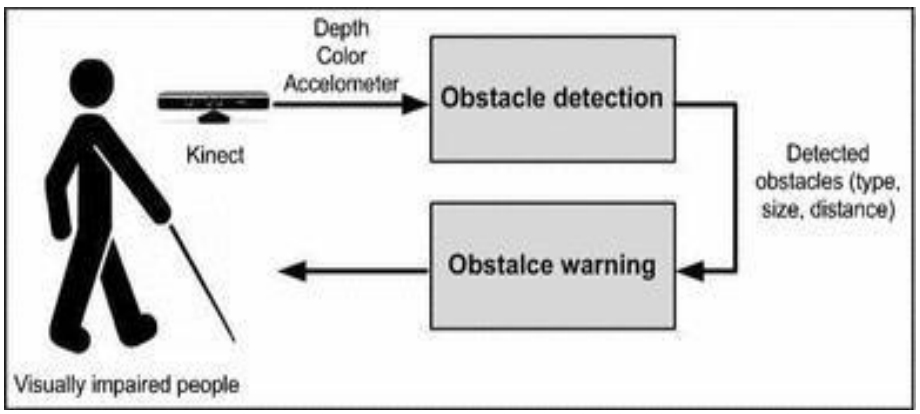


Fig 1.3(a) Object Detection Proces

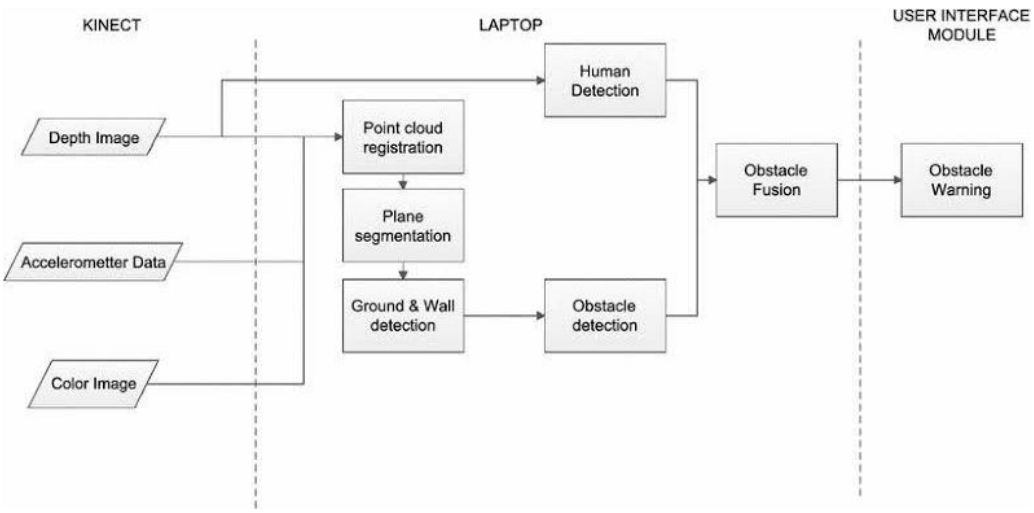


Fig 1.3 (b) Block diagram of Object Detection

1.4 Merits And Demerits

Reliable: This type of technology Provides good video quality. Difference between various objects like chair and table etc. can be easily differentiated and exact path can will be detected for visually impaired people. Scalable: This application can be run on various operating system. Object will not be stationary so it will captured the ongoing video and process all the developing steps for detection and placement of object. This feature highlights the merit. Efficient cost: The cost will be depend on the smart phones. Open Source: Android application is an open source utility command which is Linux based and released under apache software. It has many versions with extending features and properties.(e.g. lollipop, jellybean, kit Kat etc.) This application is mostly useful for blind person. No need to carry walking stick

II. OBJECTIVES AND METHODOLOGY

The object is detected using webcam and the same objects name is detected and displayed. The displayed objects name is converted from text to speech.

2.1 Methodology

Text-to-speech (TTS) is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a Web page. TTS can enable the reading of computer display information for the visually challenged person, or may simply be used to augment the reading of a text message.

Current TTS applications include voice-enabled e-mail and spoken prompts in voice response systems. TTS is often used with voice recognition programs.

Like other modules the process has got its own relevance on being interfaced with, where Raspberry Pi finds its own operations based on image processing schemes. So once image get converted to text and thereby it could be converted from text to speech. Character recognition process ends with the conversion of text to speech and it could be applied at anywhere.

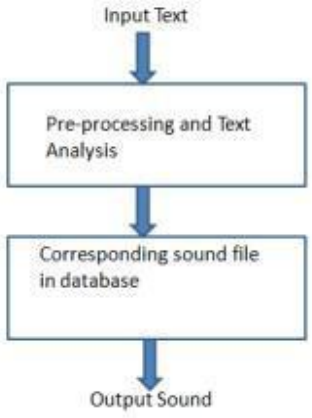


Fig 2.1 Flow Chart Diagram

Another method for converting the text into speech can be through the ASCII values of English letters. By using this method the coding length can be decreased. There are many Text to Speech converters are there but there performance depends on the fact that the output voice is how much close to the human natural voice. For example, consider a name pretty, it can be a name of a person as well as it can be defined asbeautiful. Thus it depends on how the words are pronounced.

Many text to speech engines does not give the proper pronunciation for such words thus combining some voice recordings can give more accurate result. The TTS system converts an English text into a speech signal with prosodic attributes that improve its naturalness. There are many systems which include prosodic processing and generation of synthesized control Parameters. The proposed system provides good quality of synthesized speech.

2.1(a) Yolo algorithm

There are a few different algorithms for object detection and they can be split into two groups:

1. Algorithms based on regression – instead of selecting interesting parts of an image, we're predicting classes and bounding boxes for the whole image in one run of the algorithm. Most known example of this type of algorithms is YOLO (You only look once**)** commonly used for real-time object detection.

Before we go into YOLOs details we have to know what we are going to predict. Our task is to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors:

1. centre of a bounding box (**bxby**)
2. width (**bw**)
3. height (**bh**)
4. Value **c** is corresponding to a class of an object (car, traffic lights,…).

We've got also one more predicted value pc which is a probability that there is an object in the bounding box, I will explain in a moment why do we need this.

Like I said before with YOLO algorithm we're not searching for interested regions on our image that could contain some object. Instead of that we are splitting our image into cells, typically its 19×19 grid. Each cell will be responsible for predicting 5 bounding boxes (in case there's more than one object in this cell). This will give us 1805 bounding boxes for an image and that's a really big number!

2.1(b) Working of Yolo

YOLO trains and tests on full images and directly optimizes detection performance. YOLO model has several benefits over other traditional methods of object detection like the following.

- First, YOLO is extremely fast. Since frame detection in YOLO is a regression problem there is no need of complex pipeline. We can simply run our neural network on any new image at test time to make predictions.

- Second, YOLO sees the entire image during training and testing unlike other sliding window algorithms which require multiple iterations to process a single image.

- Third, YOLO learns generalizable object representations. When trained on real time images and tested, YOLO outperforms top detection methods like DPM and R-CNN.

YOLO network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means our network reasons globally about the full image and all the objects in the image. The YOLO design enables end-to-end training and real time speeds while maintaining high average precision .

Following are the steps how YOLO works,

- First it divides the input image into an S × S grid as shown in fig.1.

Fig 2.1(a) Divide the image into S × S grid

- If the centre of an object falls into a grid cell, that grid cell is responsible for detecting that object.
- Each grid cell predicts B bounding boxes and confidence scores for those boxes as shown in fig.2.
- These confidence scores reflect how confident the model is that the box contains an object. If no object exists in that cell, the confidence scores should be zero.

Fig 2.1(b) Calculate bounding boxes and confidence score for each box.

- Each grid cell also predicts conditional class probabilities.
- These probabilities are conditioned on the grid cell containing an object. We only predict one set of class probabilities per grid cell, regardless of the number of boxes B.
- Finally, we multiply the conditional class probabilities as shown in fig.3 and the individual box confidence predictions which gives us class-specific confidence scores for each box as shown in fig.4.

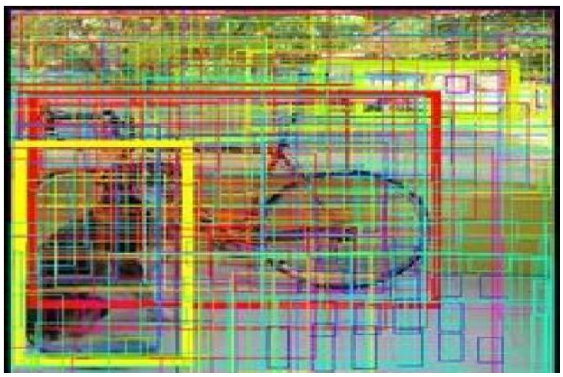


Fig 2.1(c) multiply probability and confidence scores.

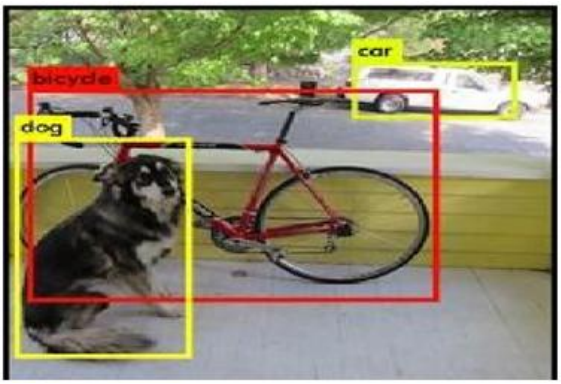


Fig 2.1 (d) Final Output

WEB SCRAPING AND TEXT TO SPEECH CONVERSION

Web scraping is a technique that is used to retrieve the content from websites. It consists of two phases namely fetching the web page and later extracting the required content from it. Here two types of web scraping is done one is extracting the content from Wikipedia and other is top google search links for that label. The required modules are installed to system using pip.

A. **Content Retrieval from Wikipedia**

After detecting the object from image will use that labelled class to retrieve data from Wikipedia. It is a free encyclopaedia in web. So by extracting data from Wikipedia helps the user to get a idea about what the object is and its uses. Wikipedia is a python library that will help to access and extract data from Wikipedia. In that module with a help of a predefined function Summary(), label(object name) and filter(no of lines from Wikipedia) are arguments for this function and returns a string that contains the extracted data.

B. **URL Retrieval from Google**

By using the label(object name) will extract top google URL's from google with the help of python module Google search. By using pre defined function called Search() will extract the required URL's. In this function we can pass arguments like label(object name) , no of links need to be extracted etc. With these links they can refer more about the object other than Wikipedia content[20].

C. **Text to Speech Conversion**

This step will convert the label(object name) and Wikipedia content to voice so that everybody can understand better. The module used for text to speech conversion is pyttsx which is platform independent and it can convert in offline too. But pyttsx is supported only in python 2.x versions so pyttsx3 module can used in both python 2.x and 3.x versions. In order to use pyttsx3 init() function need to be called to initialize the process and use a predefined method say() with argument text which needs to be converted to voice[19]. Finally use runAndWait() to run the speech.

2.2 Performance Analysis

To analyze the performance of YOLO, it compared with algorithms like R-CNN, fast R-CNN, faster R-CNN on various performance measures like time taken, accuracy and the frames per second. When analysis was done based on time taken by the algorithm to detect the objects as listed

in table 1, it is found that R-CNN takes around 40 to 50 seconds, fast R-CNN takes 2 seconds, faster R-CNN takes 0.2 seconds, and YOLO takes just 0.02 seconds. From this analysis it can be inferred that, YOLO performs 10 times quicker that faster R-CNN, 100 times quicker than fast RCNN and more than 1000 times quicker than R-CNN.

Table I: Performance Evaluation Based on Time Taken

Algorithm	Time taken (in sec)
R-CNN	40-50
Fast R-CNN	2
Faster R-CNN	0.2
YOLO	0.02

When analysis was done based on the number of frames per second, YOLO performs far better than all the other algorithms as shown in fig.5, with 48 fps whereas, R-CNN processes 2 fps, fast R-CNN processes 5 fps and faster R-CNN processes 8 fps.

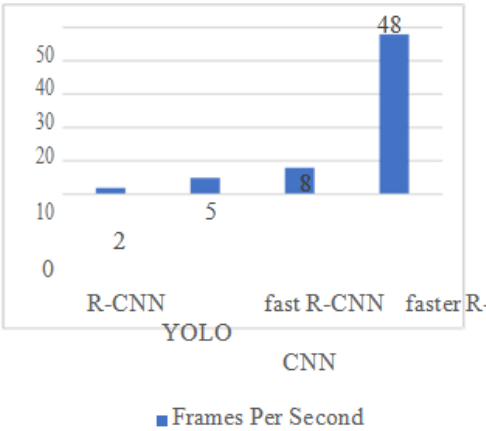


Fig.2.2(a) Performance analysis based on frames per second

When analysis was done based on the accuracy it is found that YOLO has lesser accuracy than the other three algorithms as shown in fig.6. So, it is not recommended to use YOLO for applications in which accuracy is the major concern.

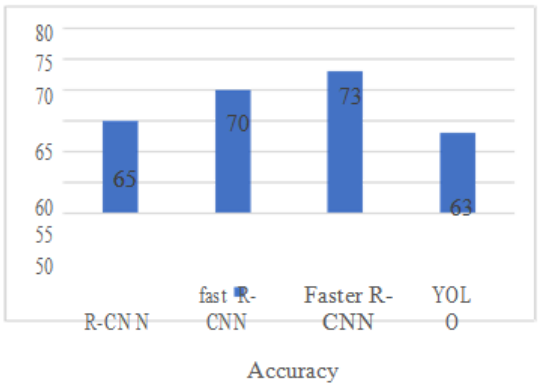


Fig. 2.2 (b) Performance analysis based on accuracy

The model can be used in tracking objects for example tracking a ball during a football match, tracking movement of a cricket bat, tracking a person in a video, Video surveillance, Smart Class for students, Instructor for blind people to get details about unknown objects. It is also used in Pedestrian detection.

2.3 Properties

- **Face detection:** An example of object detection in daily life is that when weupload a new picture in Facebook or Instagram it detects our face using this method.

- **People Counting:** Object detection can be also used for people counting, it meansthat it is used for analysing store performance or crowd statistics during festivals where the people spend a limited amount of time and other details .This type of analysis is little difficult as people move away from frame.

- **Vehicle detection:** When the object is a vehicle such as a bicycle or car or bus,object detection with tracking can prove effective in estimating the speed of the object. The type of ship entering a port can be determined by object detection based on the shape, size etc. This method of detecting ships has been developed in certain European Countries.

- **Manufacturing Industry:** Object detection is also used in industrial processes toidentify products. If we want our machine to detect products which are only circular we can use Hough circle detection transform can be used for detection

- **Online images:** Apart from these object detection can be used for classifyingimages found

online. Obscene images are usually filtered out using object detection.

- **Security:** In the future we might be able to use object detection to identifyanomalies in a scene such as bombs or explosives (by making use of a quadcopter).

- **Medical Diagnose:** Use of object detection and recognition in medical diagnose todetect the X-Ray report, brain tumours.

III. LITERATURE

3.1 Literature Survey

1. **The Cross-Depiction Problem: Computer Vision**

Algorithms for Recognising Objects in Artwork and in Photographs

The cross-depiction problem is that of recognising visual objects regardless of whether they are photographed, painted, drawn, etc. It is a potentially significant yet under-researched problem. Emulating the remarkable human ability to recognise objects in an astonishingly wide variety of depictive forms is likely to advance both the foundations and the applications of Computer Vision.

In this paper we benchmark classification, domain adaptation, and deep learning methods; demonstrating that none perform consistently well in the cross depiction problem. Given the current interest in deep learning, the fact such methods exhibit the same behaviour as all but one other method: they show a significant fall in performance over in homogeneous databases compared to their peak performance, which is always over data comprising photographs only. Rather, we find the methods that have strong models of spatial relations between parts tend to be more robust and therefore conclude that such information is important in modelling object classes regardless of appearance details.

2. **Histograms of Oriented Gradients for Human Detection**

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

3. **Speech YOLO: Detection and Localization of Speech Objects**

In this paper, we propose to apply object detection methods from the vision domain on the speech recognition domain, by treating audio fragments as objects. More specifically, we present Speech YOLO, which is inspired by the YOLO algorithm [1] for object detection in images. The goal of Speech YOLO is to localize boundaries of utterances within the input signal, and to correctly classify them. Our system is composed of a convolutional neural network, with a simple least-mean-squares loss function. We evaluated the system on several keyword spotting tasks that include corpora of read speech and spontaneous speech. Our system compares favourably with other algorithms trained for both localization and classification.

4. **Detection and Content Retrieval of Object in an Image using**

YOLO

It is easy for human beings to identify the object that is in an image. Even if the task is complex, human beings require only a minimal effort. Since computer vision is actually replicating human visual system, the same thing can be achieved in computers when they are trained with large amount of data, faster GPUs and many advanced algorithms. In general terms, Object detection can be defined as a technology that detects instances of object in images and videos by mimicking the human visual system functionalities. The motivation of the paper is making the search process easier for the user i.e., if the object is very new for the user and he has no idea about it, he can upload a picture of that object and the algorithm will detect the object and gives a description about it. The objective of the paper is to detect the object in an image, once the object is detected, the label i.e., the name of the detected object is searched in Wikipedia and few lines of description about that object is retrieved and printed. Also, the label is searched in google and the URL of the top pages with content related to the label are also displayed. The detection of object in an image is done using YOLO (You Only Look Once) algorithm with pre-trained weights. Previous methods for object detection, like R-CNN and its variations, used a

pipeline to perform this task in multiple steps. This can take some time for execution, complex optimization may beinvolved because individual training of components is required. YOLO, does it all fastly with a single neural network. Hence, YOLO is preferred.

5. **Real Time Two Way Communication Approach for Hearing Impaired and Dumb Person Based on Image Processing**

In the recent years, there has been rapid increase in the number of deaf and dumb victims due to birth defects, accidents and oral diseases. Since deaf and dumb people cannot communicate with normal person so they have to depend on some sort of visual communication. Gesture shows an expressive movement of body parts such as physical movements of head, face, arms, hand or body which convey some message. Gesture recognition is the mathematical interpretation of a human motion by a computing device. Sign language provide best communication platform for the hearing impaired and dumb person to communicate with normal person. The objective of this research is to develop a real time system for hand gesture recognition which recognize hand gestures, features of hands such as peak calculation and angle calculation and then convert gesture images into voice and vice versa. To implement this system we use a simple night vision web-cam with 20 megapixel intensity. The ideas consisted of designing and implement a system using artificial intelligence, image processing and data mining concepts to take input as hand gestures and generate recognizable outputs in the form of text and voice with 91% accuracy.

3.2 Case Study

Recently automatic speech recognition (ASR) has became ubiquitous in many applications. While ASR systems like *DeepSpeech 2* and *wav2letter* reached amazing results in transcribing read and conversational speech, sometimes it is desired to spot and locate a predefined small set of words with extremely high accuracy. For example, services like *Google Now* or *Apple's Siri* can be activated by pronouncing "OK Google" or "Hey Siri", respectively. It is also used by intelligence services to accurately find specific keywords while monitoring suspected phone calls. The task of detecting and localizing words can be used to automatically analyze the diadochokinetic articulatory task that is used to analyze pathological speech and hence cannot be performed effectively with ASR systems. In this work we present an end-to-end system that goes from a speech signal to the transcription and alignment of given keywords (this is in contrast to the spoken term detection task that makes predictions on keywords that it has not been trained on).

Our architecture performs both detection and localization of these predefined keywords. Previous works typically focus on only one of these two challenges. Namely, algorithms would either predict what words appear in a given utterance, thus performing detection, or are given the audio signal and the target transcription and align them, thus performing localization mostly using forced alignment [8, 9]. Keshet *et al.* proposed to use the confidence of a phoneme aligner and anexhaustive search to detect and localize terms that are given by their phonetic content.

In the vision domain, object detection algorithms combine the two aforementioned tasks: detection of the desired object and its localization in the image. Specifically, the YOLO and SSD algorithms identify objects in an image using bounding boxes. Inspired by the idea of using bounding boxes for object detection in images, we propose to identify speech objects in an audio signal. More specifically, consider the word classification task as a form of object detection for a speech signal.

Palaz *et al.* presented work that is most similar to ours. Their algorithm was trained to jointly locate and classify words. However, they used a weakly supervised setting and did not use word alignments, and hence were unable to perfectly predict the whole time-span of the predicted words. In our work, however, our goal is both to detect and to locate the entire span of every word, so both tasks' results are strongly accurate.

This paper is organized as follows. In Section 2 we formally introduce the classification and localization problem setting. We present our proposed method in Section 3, and in Section 4 we show experimental results and various applications of our derived method. Finally, concluding remarks and future directions are discussed in Section 5.

3.3 Problem Setting

The input to our system is a speech utterance. The input speech utterance is represented as a series of acoustic feature vectors. Formally, let $\mathbf{x}^{-}$ = ($\mathbf{x}\in1,...,\mathbf{x}T$) denote

the input speech utterance of a fixed duration T, where each $\mathbf{x}t$ R^D is a D dimensional vector for $0 \leq t \leq T$. We further define the lexicon Ł = $\{k1,k2,...,kL\}$ to be the target set of L keywords or terms that may appear in the audio signal $\mathbf{x}^{-}$. Note that the utterance does not necessarily contain any of these keywords or may contain several of them. In our setting the speech objects are the L

keywords, but the model proposed here is not limited to specific keywords and can be used to detect and localize any audio or speech object, e.g., the syllables */pa/*, */ta/*, and */ka/* in the diadochokinetic articulatory task. Our goal is to spot all the occurrences of the keywords in a given utterance $\mathbf{x}^{-}$ and estimate their corresponding locations.

We assume that *N* keywords were pronounced in the utterance $\mathbf{x}^{-}$, where $N \geq 0$. Each of these *N* events is defined by its lexical content and its time location∈. Each

such event *e* is defined formally by the the tuple $e = (k, t^k_{\text{start}}, t^k_{\text{end}})$, where *k* Ł is the

actual keyword that was pronounced, and t^kstart and t^k_{end} are its start and end times, respectively. Our goal is therefore to find all the events in an utterance, so that for each event the correct object *k* is identified as well as its beginning and end times.

3.4 Model

As previously mentioned, our model is inspired by the YOLO model . We now describe our model formally. Our notation is schematically depicted in Fig. 1. We assume that the input utterance $\mathbf{x}^{-}$ is of a fixed size *T* (1 second in our setting). We divide the input-time to *C* non-overlapping equal sections called *cells* (*C* = 6 in our setting). Each cell is in charge of detecting a single event (at most) in its time-span. That is, the *i*-th cell, denoted *ci*, is in charge of the portion [*tci*,*tci*+1 − 1], where

tci is the start-time of the cell and *tci*+1− 1 is its end-time, for

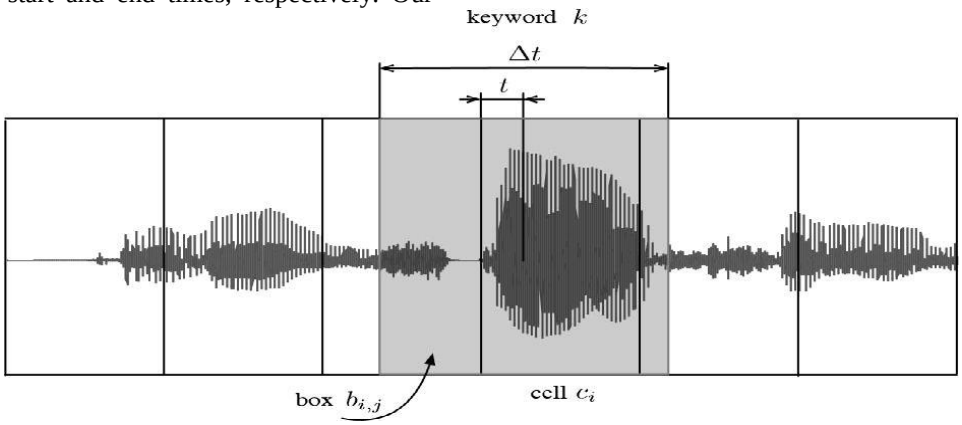


Fig 3.4 Vocal Representation

Figure 3.4: The notation used in our paper. The keyword "star" is found within cell *ci*. One of the timing boxes *bi,j* is depicted with a shaded box, and it defines the timing of the keyword relative to the cell's boundaries.

$1 \leq i \leq C$. The cell estimates the probability Pr(*k*|*ci*) of each keyword *k*∈ Ł to be uttered within its time-span. We denote the estimation of this probability by *p*ci(*k*).

The cell is also in charge of localizing the detected event. The localization is defined relative to the∈ cell's boundaries. Specifically, the location of the event is defined by the time ^{t} [*tci*,*tci*+1− 1], which is the center of the event relative to the cell's boundaries, and Δ*t*, the duration of the event. Note that Δ*t* can be longer than the time-span of the cell. Using this notation the event spans [*tci* + *t* − Δ*t*/2,*tci* + *t* + Δ*t*/2].

In order to localize effectively, each cell is associated with *B timing boxes* (called *bounding boxes* in the YOLO literature). Each box *bi,j* of the cell *ci* tries to independently localize the event and estimate the probability of the timing given the presumed keyword, Pr(*t*,Δ*t*|*k*). It is defined by the tuple (*tj*,Δ*tj*,*p*b*i,j*), where *p*b*i,j* is the confidence score of the localization *t*,Δ*t* and it can be considered as an estimation of the probability Pr(*t*,Δ*t*|*k*,*ci*).

We now turn to describe the model's inference. The inference for each cell is performed independently. For the *i*-th cell, *ci*, the chosen event is composed of the keyword *k*ˆ and the timing *t*,ˆΔˆ*t* that maximizes the conditional probabilities.

Namely, $(k', t', \Delta t') = \arg\max_{k,t,\Delta t} \Pr(k, t, \Delta t | c_i)$ (1)

= arg max Pr(*k*|*ci*) Pr(t,Δt|*k*,*ci*). (2) *k*,*t*,Δ*t*

The first conditional probability in Eq. (2) is *pci*(*k*), whereas the second conditional probability is *p*b*i,j* of box *bi,j*. Since there are *L* keywords and *B* boxes

the search space reduces to $L \times B$ elements, hence it is very efficient:

$$\max_{k \in Ł} \max_{1 \le j \le B} p_{c_i}(k)\, p_{b_{i,j}}.$$

Finally, the event is considered to exist in the cell if its conditional probability from above is greater than a threshold, θ.

We conclude this section by describing the training procedure. Our model, *SpeechYOLO*, is implemented as a convolutional neural network. The initialconvolution layers of the network extract features from the utterance while fully connected layers are later added to predict the output probabilities and coordinates. Our network architecture is inspired by PyTorch's implementation of the VGG19 architecture[1], and is presented in Section 4.

The training set is compose of examples, where each example is an event that is

composed of the tuple $(\bar{\mathbf{x}}, k, t^k_{\text{start}}, t^k_{\text{end}})$.

We denote by $1^k i$ the indicator that is 1 if the keyword k was uttered within the cell ci, and 0 otherwise. Formally,

$$\mathbb{1}_i^k = \begin{cases} 1 & t^k_{\text{start}} \ge t_{c_i} \ \wedge \ t^k_{\text{end}} \le t_{c_{i+1}} - 1 \\ 0 & \text{otherwise} \end{cases}$$

When we would like to indicate that the keyword is not in the cell we will use the notation $(1 - 1^k i)$.

The model's loss function is defined as a sum over several terms, each of which took into consideration a different aspect of the model, as a follows:

$$\bar{\ell}(\bar{\mathbf{x}}, k, t^k_{\text{start}}, t^k_{\text{end}}) = \lambda_1 \sum_{i}^{C} \sum_{j}^{B} \mathbb{1}_i^k \left(t_j - t'_j\right)_2$$

$$= + \lambda_2 \sum_{i=1}^{C} \sum_{j=1}^{B} \mathbb{1}_i^k \Big| + \sum_{i=1}^{C} \sum_{j=1}^{B} \mathbb{1} \quad \sum \mathbb{1}_i^k \big(1 - + \lambda_3 \sum_{i=1}^{C} \sum_{j=1}^{B} \big(1 \quad =1\ k \in Ł$$

We would like to note that our system is inspired by the first version of YOLO. Further research on YOLO has been conducted in. It seems, however, that most expansions made to their algorithm are irrelevant to for our domain. In the authors' main contributions are the addition of *anchor boxes*, which defines constraints on the shapes of the bounding boxes. This lead to specifying a separate class probability value for every bounding box. This is relevant when dealing with a multidimensional domain, and is less relevant for speech. In their paper, they additionally suggest the usage of a fully convolutional network, i.e. replacing the fully connected layers with convolutions. We found that this yielded inferior results for our dataset. In , the main development was the shift from multiclass classification to multilabel classification. This changed the loss function from using regression to using cross entropy instead. This too is irrelevant for our domain.

3.5 Experiment

We used data from the LibriSpeech corpus , which was derived from read audio books. The training set consisted of 960 hours of speech. This corpus had two test sets: *test clean* and *test other*, which summed up to 5 hours of speech. The first set wascomposed of high quality utterances and the second set was composed of lower quality utterances. The audio files were aligned to their given transcriptions using the Montreal Forced Aligner (MFA) [9]. We extracted the Short-Time Fourier Transform (STFT) as features to the sound files using the librosa package. These features were computed on a 20 msec window, with a shift of 10 msec.

A target event of the input speech signal can be defined as any discrete part of an utterance that is discernible to a human annotator. Hence, events could be defined as a set of words, phrases, phones, etc. It was assumed that only events from the selected lexicon Ł are available during training time.

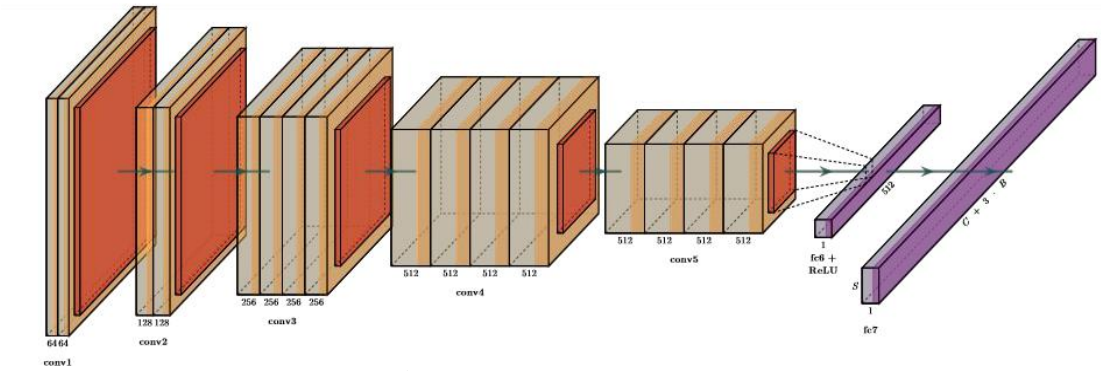


Fig 3.5 Detection N/W Layers

Figure 3.5: The detection network has 16 convolutional layers followed by 2 fully connected layers. Every convolutional layer is followed by BatchNorm and ReLU. We pretrained the convolutional layers on the Google Command classification task and then replace the final layer for detection and localization.

We used a convolutional neural network that is similar to the VGG19 architecture. It had 16 convolutional layers and 2 fully connected layers, and the

final layer predicted both class probabilities$_*$ and timing boxes' coordinates. We denote this architecture as *V GG*19 . The model is described in detail in Figure 2.

For comparison,$_*$ we also implemented a version of the VGG11 model (denoted by *VGG*11), which had less convolutional layers. Both models were trained using Adamand a learning rate of 10^{-3}. We pretrained our convolutional network using the Google Command dataset for $L = 30$. We later replaced the last linear layer in order to perform prediction on a different number of events.

We divided our experiments into two parts: in the first, we evaluated SpeechYOLO's capability to correctly predict and localize words within an utterance, and compared its performance to other similar systems. In the second part, we evaluated SpeechYOLO for the keyword spotting task on various domains.

Word prediction and localization

In this subsection, we evaluated the system's capability to learn word detection and localization. We defined the target events to be the 1000 most common words in the training set (L = 1000). It turned out that the average utterance time of a single word in our corpus was approximately 0.2 seconds. To assure that the timing cells properly covered the span of the speech signal, we chose to use $C = 6$ timing cells per utterance of $T = 1$ sec. We arbitrarily set the number of timing boxes per cell to be $B = 2$.

We chose the value of the threshold θ that maximizes the F1 score, which is defined as the harmonic mean of the precision and recall. We evaluated the model's detection capabilities using Precision and Recall. Results are presented in Table 1. It seems that the proposed system was able to correctly detect most of the words, with *V GG*19$*$outperforming *V GG*11$*$, due to its size and enhanced expressiveabilities.SpeechYOLO evaluations with two architectures on both of LibriSpeech's

Prediction and Localization

Due to the uniqueness of our system's aim to both classify and localize words, it is challenging to find an equivalent algorithm to justly compare with. Most other algorithms focus on either one of the tasks, but not on both. The system of Palaz, Synnaeve and Collobert [12] was developed for weakly-supervised word recognition; that is, its aim is to perform word classification and find word position, while training with a BoW supervision. As in [21], we refer to this system as PSC.

PSC receives the Mel Filterbanks coefficients as input features. Their architecture is composed of 10 convolutional layers. The final convolution has 1000 output filters for very time span, with every filter corresponding to a word k in the lexicon Ł. The idea is that the score for word k would be highest in the time span it occurred in. The system is trained using SGD with a learning rate of 10^{-5}.

We compared SpeechYOLO's prediction and localization abilities to PSC's, as shown in Table 2. We calculated the F1 measure as before. The *Actual* accuracy measure was calculated as described in [12], and measures localization as well

as prediction. For PSC, the *Actual* accuracy was calculated as follows: word detection was performed by thresholding the probability of a word being present in the sequence. For every word *k* that passed the chosen threshold, we chose the frame in which it received the highest score. We then assessed if this frame was located within the range of *k* stated by the ground truth alignment. The closest equivalent of this measure for our model was to choose this frame to be the center of the predicted timing box. This value was in turn compared to the ground truth alignment. As before, the threshold θ was chosen to maximize the F1 measure. SpeechYOLO clearly outperformed PSC for both the F1 score and the Actual accuracy measure.

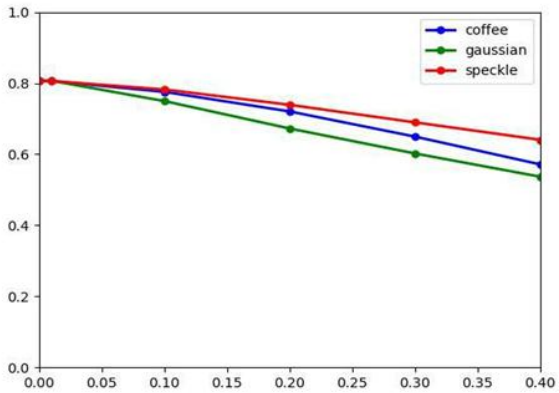
To assess the strength of SpeechYOLO's localization ability, we calculated both systems' average intersection over union (IOU) value with the ground truth alignments. While SpeechYOLO's IOU value clearly outperformed that of PSC, one must remember that PSC was not trained with aligned data.

Table 2: Comparing SpeechYOLO and PSC [12]'s evaluations of the F1 score, Actual accuracy and average IOU value. The threshold value that maximized the F1 score was chosen (θ = 0.4).

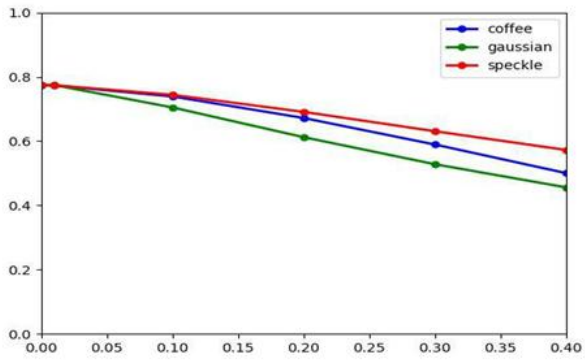
	F1	Actual	IOU
SpeechYOLO	0.807	0.774	0.843
PSC	0.767	0.692	0.3

We further checked the quality of SpeechYOLO's localization capability. To do so, we compared SpeechYOLO with MFA, after both had been trained on LibriSpeech. We tested them on the training set of the TIMIT corpus. TIMIT is a corpus of read speech, and presents a different linguistic context compared to LibriSpeech. The IOU measure was used to compare both algorithm's output alignments with TIMIT's given word alignments for the 1000 most common words in the LibriSpeech training set. In order to predict SpeechYOLO's IOU values, it was assumed that its predictions were perfect. This was due to the fact that SpeechYOLO does not receive transcriptions as an input, and because our goal was to asses the localization task alone. The IOU of SpeechYOLO on TIMIT was 0.673, while MFA achieved 0.827.

The forced aligner, MFA, performs its alignments using a



(a) F1 score



(b) Actual accuracy

Figure 3: SpeechYOLO's performances when injecting background noise. The y-axis is the measure and the x-axis is the strength of the noise added (α).

complete transcription of the words uttered in a speech signal. On the other hand, SpeechYOLO receives no information about the words uttered. Hence, given MFA's extended knowledge, we considered its localization ability as an "upper bound" to ours. Therefore, we found that SpeechYOLO's IOU value, while lower than MFA's, were sufficiently high.

Additionally, an aligner could naturally go wrong if there are incorrect or missing words in its transcription, or alternatively if the audio signal contains long silences or untrascripted noises between words (e.g. a laugh or a cough) [22]. It should be noted that given SpeechYOLO's lack of knowledge about the transcription, these problems do not affect it.

Robustness to noise

We further demonstrated SpeechYOLO's robustness by artificially adding background noise to LibriSpeech's audio files with a relative amplitude α. We injected 3 types of background noises: a coffee shop, gaussian noise, and speckle. In Figure 3 we show SpeechYOLO's F1 score and

Actual accuracy measures when increasing the α variable, thus intensifying the injected noise. It is apparent that minor amounts of noise do not degrade SpeechYOLO's performances. Note that SpeechYOLO was able to deal even with higher α values, although it yielded somewhat reduced results.

Keyword spotting

In this part, we evaluate SpeechYOLO on a real-world application: keyword spotting. For evaluation, we use the F1 metric, and the Maximum Term Weight Value (MTWV) metric [23].

MTWV is defined as one minus the weighted sum of the probabilities of miss and false alarm.

LibriSpeech Corpus

We compare SpeechYOLO's keyword spotting capabilities with those of the PSC system. In their work, they use a set of keywords that is a subset of the 1000 words used previously for prediction and localization. The chosen keywords are in Table 2 of [12], and are evaluated on both of LibriSpeech's test sets. A comparison of our results are presented in Table 3. Here too SpeechYOLO's results outperformed those of PSC.

Table 3: MTWV values for SpeechYOLO and PSC on the keyword spotting task, evaluated on both of LibriSpeech's test sets.

	SpeechYOLO	PSC
test clean	0.74	0.72
test other	0.38	0.27

Spontaneous speech corpus

We now present the results of SpeechYOLO for keyword spotting with spontaneous speech. This is relevant for mobile applications, where a device is activated by a voice command like 'OK Google" or "Hey Siri". To simulate this task, we use a corpus taken from a daily TV show *Good evening with Guy Pines*[2]. This corpus, which we will call "Hi Guy", consists of spontaneous and noisy recordings. In each recording, a celebrity is prompted to utter the phrase *Hi Guy!* These recordings vary greatly in terms of their environment and the speakers within them are highly diverse.

The corpus consists of 880 examples, out of which 445 contain the chosen keyword. We chose the phrase "Hi Guy" to be the keyword that our system searches for. The input length is 3 seconds. The system achieves an *Actual* accuracy of 0.624, and an F1 score of 0.755 (precision: 0.748, recall: 0.761). We find these results to be surprisingly satisfying due to the small size of the dataset and due to the diversity found in the corpus: the audio files are at times extremely noisy, the pronunciation of the speakers vary, and the keyword is sometimes sung instead of being spoken.

REQUIREMENTS SPECIFICATION

H/W System Configuration:-

- RAM - 8GB (min)
- Hard Disk - 2 GB
- Floppy Drive - 1.44 MB

- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor

Software requirements

Python 2.7 or higher

- Pycharm
- Openscv
- Window-8,10

FEASIBILITY STUDY

We frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. In this pattern we are using a Yolo module by using object detection purpose and it will show boundary of on object. and display text and speech well also come.

ADVANTAGES

- Yolo is very much faster than all another object detection algorithm.
- Excellent balanced speed and accuracy.

DISADVANTAGES

- It can't identify the small objects in the image.
- The each grid cells only predicts two boxes and can only have one class.

EXISTING SYSTEM

We frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding and class probabilities directly from full images in one evaluation.

PROPOSED SYSTEM

YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the MAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict

false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and CNN, when generalizing from nature images to other domains like network TEM.

STATUS AND ROADMAP

A. Testing data

Available video databases consist generally on videos of traffic vehicles, faces detection and tracking. In our work we need videos of scenes from daily life with images of all the objects. We precede with four videos sequences each one from them contains a number of daily life objects. Videos have not the same duration. Table I shows that each one contains different number of frames and objects. They will have different time processing.

B. Object detection and identification in video

The first step is to load up a video scene containing objects; then we have the feature extraction. We use the difference-of Gaussian feature detector introduced previously with SIFT descriptor. The features we find are described in a way which makes them invariant to size changes, rotation and position. These are quite powerful features and are used in a variety of tasks. We use a standard Java implementation of SIFT. The SIFT descriptor is a 128 dimensional description of a patch of pixels around the key point. In the step of matching, the basic matcher finds many matches, many of which are clearly incorrect. Number of scales True positive (%) False positive (%) 3 35 65 5 95 5 An algorithm called Random Sample Consensus (RANSAC) [16] is used to fit a geometric model called an affine transform to the initial set of matches. This is achieved by iteratively selecting a random set of matches, learning a model from this random set and then testing the remaining matches against the learnt model. We can take advantage of this by transforming the bounding box of the object with the transform estimated in the affine transform model. Therefore we can draw a polygon around the estimated location of the object within the frame. Then, Table II shows that the number of true positive depends on the number of scales used in the SIFT algorithm. In fact, 5 scales gives a better matching than 3, also a number up to 5 didn't' give a butter results but it takes more time. So, in order to have strong matches, we work with number of scale S equal to 5. In the figure we tried to find some object. We note that we detect all objects in this scene video. So we tried also to detect a medical box, because we know that identifying medicines is a delicate task for blind people. Medical box detection in video with high luminosity. The box is well detect, a short description about the medicine in an audio file notify the blind about what he hold in his hand. In the video scene we can have different level of illumination. Even in the same video, illumination can be changed. It was important to detect objects in video scene with high illumination and in another one with low illumination It is clear that the number of true matches decrease in the video with low lightness. However,the object is well detected. So we can conclude that SIFT is invariant to the change in luminosity in video and the object can be detected and identified.

C. Discussions

The challenge in comparing key points is to figure out matching between key points from some frames and those from target objects. We get high percentage of detected objects but we tried also to identify the reason behind some failure cases. The first cause of non-detection of object was the quality of images.

IV. CONCLUSION

We introduce YOLO, a unified model for object detection. Our model is simple to construct and can be trained directly on full images. Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly. Fast YOLO is the fastest general-purpose object detect-tor in the literature and YOLO pushes the state-of-the-art in real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection and text to speech is also done.

REFERENCES

[1]. Shinde, Shweta S., Rajesh M. Autee, and Vitthal K. Bhosale. "Real time two way communication approach for hearing impaired and dumb person based on image processing." Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on. IEEE, 2016.

[2]. Shangeetha, R. K., V. Valliammai, and S. Padmavathi. "Computer vision based approach for Indian Sign Language character recognition." Machine Vision and Image Processing (MVIP), 2012 International Conference on. IEEE, 2012.

[3]. Sood, Anchal, and Anju Mishra. "AAWAAZ: A communication system for deaf and dumb." Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2016 5th International Conference on. IEEE, 2016.

[4]. Ahire, Prashant G., et al. "Two Way Communicator between Deaf and Dumb People and Normal People." Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE, 2015.
[5]. Ms R. Vinitha and Ms A. Theerthana. "Design And Development Of Hand GestureRecognition System For Speech Impaired People."
[6]. Kumari, Sonal, and Suman K. Mitra. "Human action recognition using DFT." Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2011 Third National Conference on. IEEE, 2011
[7]. Vanitha E, Kasarla PK, Kuamarswamy E. Implementation of text- to-speech for real time embedded system using Raspberry Pi processor. International Journal and Magazine of Engineering Technology Management and Research. 2015 Jul:1995.

Punyaslok Sarkar, et. al. "Object Recognition with Text and Vocal Representation." *International Journal of Engineering Research and Applications (IJERA*), vol.10 (05), 2020, pp 63-77.