Received February 4, 2020, accepted February 27, 2020, date of publication March 12, 2020, date of current version March 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980452

Towards Deep Object Detection Techniques for Phoneme Recognition

MOHAMMED ALGABRI[1,3], HASSAN MATHKOUR[1,3], MOHAMED ABDELKADER BENCHERIF[2,3], MANSOUR ALSULAIMAN[2,3], AND MOHAMED AMINE MEKHTICHE[2,3]

[1]Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[2]Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[3]Centre of Smart Robotics Research (CS2R), King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Mohammed Algabri (malgabri@ksu.edu.sa)

This work was supported by the Deanship of Scientific Research at King Saud University through the Research Group under Grant RG-1437-018.

ABSTRACT The use of cutting edge object detection techniques to build an accurate phoneme sequence recognition system for English and Arabic languages is investigated in this study. Recently, numerous techniques have been proposed for object detection in daily life applications using deep learning. In this paper, we propose the use of object detection techniques in speech processing tasks. We selected two state-of-the-art object detectors, namely YOLO and CenterNet, based on a trade-off between detection accuracy and speed. We tackled the problem of phoneme sequence recognition using three systems: the domain transfer learning system (DTS) from image to speech, intra-language transfer leaning system (IaTS) between speech corpora within the same language (English to English), and inter-language transfer learning system (IeTS) between speech corpora from dissimilar languages (English to Arabic). For English phoneme recognition, the Texas Instruments/Massachusetts Institute of Technology (TIMIT) corpus is used to evaluate the performance of the proposed systems. Our IaTS based on the CenterNet detector achieves the best results using the test core set of TIMIT with 15.89% phone error rate (PER). For Arabic phoneme recognition, the best performance, with 7.58% PER, was achieved using the CenterNet. These results show the effectiveness of using object detection techniques in phoneme recognition tasks. Furthermore, based on the findings of this study, speech processing tasks may be treated as object detection tasks.

INDEX TERMS CenterNet, object detection, phoneme recognition, transfer learning, YOLO.

I. INTRODUCTION

Phoneme recognition plays a dominant part in many applications such as speech recognition [1], speaker recognition [2], and pronunciation error detection and correction [3]. With the success of deep learning techniques for computer vision, many studies have been conducted on speech processing tasks by converting speech signals to a visual representation such as spectrogram [4]. Recently, research efforts have focused on object detection, which aims to localize the objects on an image and identify the class of each object [5]. In this paper, we consider phonemes with their time boundaries as objects in a spectrogram (i.e., image). Hence, we apply object detection techniques to detect the phonemes in the spectrogram.

In general, object detection techniques can be classified into two major classes: two-stage detectors and one-stage detector [6]. The detection process of the two-stage detectors consists of two steps. In the first step, the image is divided into candidate regions; then, each region is classified separately. An example of a two-stage detector is a region-based CNN (RCNN) [7]. In one stage detectors, the whole image is fed to the network at once to detect the objects and their respective bounding boxes. Examples of one-stage detectors are YOLO [8] and CenterNet [9]. In general, one stage detectors are faster than two-stage detectors [10]. Owing to their speeds, modernity and accuracies, YOLO and CenterNet were investigated in this study for phoneme sequence

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

recognition. The speed of the techniques is crucial for real time applications such as speech recognition. Transfer learning techniques have been applied in many domains such as image classification [11], semantic segmentation [12], and object detection [13] to overcome the lack/imbalance of training data and accelerate the training process. The aim of this study is to answer the following research questions:

Q1: What is the effectiveness of applying object detectors to phoneme recognition?

Q2: Does intra language transfer learning improve the recognition accuracy in this type of task?

Q3: Does inter language transfer learning improve the recognition accuracy in this type of task?

Q4: Can we relate the evaluation metrics of object detection to the metrics of phoneme recognition?

Our investigation is based on using English and Arabic corpora. To answer the first research question, we proposed the domain transfer learning system (DTS) by adapting YOLO and CenterNet for the phoneme recognition task using the Texas Instruments/Massachusetts Institute of Technology (TIMIT) corpus, which is in the English language. In this system, Image-Net pre-trained weights, rather than randomly initializing weights, were used as the initial weights for the backbone networks of the detectors. To answer the second research question, the intra-language transfer learning system (IaTS) is proposed. TIMIT is a small vocabulary corpus compared to other object detection benchmarks. Hence, in IaTS, we start by training the detectors of DTS using a random subset from the LibriSpeech corpus [14], which is a large English language vocabulary corpus. Subsequently, we fine-tune and test the detectors using TIMIT. We selected LibriSpeech because it is a free corpus, and its acoustic model and lexicon are freely available. To answer the third research question, we proposed the inter-language transfer learning system (IeTS). This system aims to study the effect of transfer learning between English language as a source language and the Arabic language as a target language, by fine-tuning the trained detectors of DTS by a small Arabic corpus. Finally, to answer the fourth research question, we used two evaluation metrics to evaluate the performance of the proposed methods. The first metric, mean average precision (mAP), is from the object detection domain and the second metric, Phone Error Rate (PER), is from speech recognition domain.

To the best of our knowledge, this is the first attempt to use object detection techniques for phonemes sequence recognition, except for one study on keyword spotting using the first version of YOLO [15]. That study was published few months ago during the development of our study. The rest of this paper is organized as follows: the related studies are presented in section II, a detailed research methodology is explained in section III, and experimental results and analysis are presented in section IV. Finally, the conclusions are given in section V.

II. RELATED WORK

As the topic of this study belongs to the object detection and speech recognition fields, we shall first introduce state-of-the-art techniques in object detection by exploring their growth curve from the pre-deep learning era to the present day and will highlight the most well-known benchmarks. Then, we shall present diverse related studies on TIMIT phoneme recognition. Finally, some prior studies of Arabic Automatic Speech Recognition (AASR) will be discussed.

A. OBJECT DETECTION

Object detection plays a vital role in many real life applications, such as face detection and pedestrian detection in the security field, autonomous driving and traffic sign recognition in the transportation field, flyer detection and topographic survey in the military field, glaucoma detection and skin lesion analysis in the medical filed, etc. [5]. The following section is mostly inspired from the recent comprehensive surveys in object detection techniques [5], [6], [10]. Before the deep learning era, some object detection techniques were proposed such as Viola Jones detectors [16], the histograms of oriented gradients detector [17], and discriminatively trained, multiscale, deformable part model [18], which are based on hand crafted features. During the evolution of deep learning and GPU computation, the curve of object detection techniques has grown rapidly. The deep detectors can be classified into two categories: one-stage and two-stage detectors. In two-stage detectors, an image is first divided into candidate regions and then, each region is fed to a convolutional neural network (CNN) to extract features, which are fed to detection layers to identify the class of these regions. The R-CNN [7], Fast-RCNN [19], Faster-RCNN [20], SPPNet [21], and Mask-RCNN [22] are examples of two-stage detectors. On the other hand, one-stage detectors work by feeding the entire image to a CNN to extract features, which are then fed to detection layers to predict objects and bounding boxes. YOLO and its varieties [8], [23], [24], SSD [25], RetinaNet [26], CornerNet [27], and CenterNet [9], [28] are examples of one-stage detectors. In terms of benchmarks, there are different object detection benchmarks. Pascal VOC 2007 and VOC2012 (20 classes) [29], [30], ILSVRC (200 classes) [31], MS-COCO (80 classes) [32], and OID (600 classes) [33] are examples of these benchmarks.

Most of the object detection techniques that are based on deep learning consist of two networks, backbone networks and detection networks [5]. The backbone network is used to extract the features, and the detection network is used to classify the objects and detect the corresponding bounding boxes. The backbone networks are based on a CNN without fully connected layers. In the literature, several backbones are used, such as AlexNet [34], VGG [35], ResNet [36], Darknet-53 [24], deep layer aggregation (DLA) [37], MobileNet [38], and Hourglass [39].

B. ENGLISH ASR

ASR systems have received considerable attention in the age of deep learning. Several speech corpora have been released with small and large vocabularies such as TIMIT [40], LibriSpeech [14], TEDLIUM [41], WSJ [42], and Vystadial [43]. TIMIT is a well-known, time-aligned (at word and phone level) corpus that was designed to develop ASR [44]. TIMIT is a suitable choice for investigating a new method for speech recognition because of its extensive publication, simplicity, and size [45]. Hence, we will briefly discuss the performance of ASR using the TIMIT corpus. Diverse deep learning architectures have been published for English ASR systems, such as deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN), and attention-based RNN, as described in detail in the following sections.

1) ENGLISH DNN-BASED ASR

For many decades, conventional ASRs have been developed using the Gaussian mixture model-hidden Markov model (GMM-HMM) acoustic model. ASRs based on DNNs have outperformed those based on the GMM-HMM model for different speech benchmarks [45]. An acoustic model for phone recognition based on the deep belief network (DBN) is proposed in [46]. It outperformed the other conventional approaches in terms of phone error rate by achieving a 23.0% PER on TIMIT. A DBN with multiple hidden layers was also proposed by the same authors in [47] and achieved a 20.7% PER on TIMIT. Recently, a DNN acoustic model for TIMIT phone recognition based on multi resolution speech representation proposed in [48] achieved the best PER of 18.25%. The performances of a feed forward DNN, time delay neural network (TDNN), and long short-term memory (LSTM) are explored in [44] for TIMIT phone recognition, where LSTM-based phone recognition achieved a PER of 15.02%.

2) ENGLISH CNN BASED ASR

Abdel-Hamid *et al.* [49] proposed a CNN acoustic model for TIMIT phone recognition and large vocabulary speech recognition. They compared the performances of an acoustic model based on CNN and an acoustic model based on DNN, and noticed that the model based on the CNN outperformed that based on the DNN. For TIMIT phone recognition, they reported 21.87% and 20.17% as an average PERs for the DNN and CNN-based models, respectively. A hierarchical CNN based on the maxout activation function instead of ReLU is proposed in [50] for phone recognition. A PER of 16.5% was achieved on the TIMIT core test set, which the authors claimed was the best result that had been achieved on TIMIT to that date. By replacing the softmax layer in a CNN by a support vector machine (SVM), Passricha and Aggarwal [51] developed a convolutional SVM (CSVM) for speech recognition. They experimented with different training criterion on TIMIT and reported an overall PER of 16.9%. SincNet is a novel CNN proposed for speech and speaker recognition [52]. It achieved a PER of 17.2% for TIMIT phone recognition, outperforming the standard CNN. Quaternion values based on a CNN rather than real values was proposed for TIMIT phone recognition and a PER of 19.64% was realized [53].

3) ENGLISH RNN BASED ASR

A hybrid model CNN-RNN is proposed for phone recognition and emotion recognition, and accomplished a PER of 18.0% on a TIMIT core test set [54]. Light gated recurrent units (Li-GRU) have been proposed for speech recognition [55]. Experiments were conducted using TIMIT and other speech corpora, and a PER of 14.9% was attained using feature-space maximum likelihood linear regression (fMLLR). An attention-based RNN was applied for a TIMIT phone recognition task and competitive PER of 17.6% was achieved [1]. The performance of a combination of Li-GRU and multi-layer perceptron (MLP) with concatenation of different features such as mel frequency cepstral coefficient (MFCC), filter banks (FBANK), and fMLLR using pytorch-kaldi speech recognition toolkit is presented in [56]. A PER of 13.8% for the TIMIT test set was achieved, which were the best published results at that time, as mentioned by authors.

C. ARABIC ASR

Challenges in building Arabic ASR are presented in [57]. An Arabic ASR system for broadcast news transcription is described in [58]. The experiment was conducted on an Arabic broadcast news corpus containing 7.5 h of recordings; 7 h were used for training and 0.5 h were used for testing. The reported word error rate (WER) was 10.14% on a test set of 3585 words and 8.61% for non-vocalized text (text without diacritical marks). A rule-based method for creating a pronunciation dictionary for large vocabulary Arabic ASR was proposed in [59]. The proposed method was evaluated on a broadcast news corpus with 5.4 h of recordings (4.3 for training and other remaining for testing). They reported a WER of 9% using text without diacritical marks. Building a language model for Holy Quran ASR using CMU Sphinx 4 is investigated in [60]. The authors of [61] proposed spectro-temporal directional derivative (STDD) feature for high performance Arabic ASR for a serious game. Ten digits from (1 to 10) were recorded from 50 normal and 70 pathological speakers. They used the HMM for the acoustic model and obtained a word accuracy of 99.01%.

The above studies were performed at the word level. As examples of studies on the Arabic language at the phoneme level, we present the following. A study on Arabic speech processing that concentrated on five Arabic consonants phonemes (pharyngeal and uvular), which appear only in the Arabic language, is presented in [62]. The authors conducted experiments on the WestPoint Modern Standard Arabic corpus using the HTK toolkit. They also studied the effect of non-native speakers on the recognition rate of these phonemes. The highest recognition rates were for the pharyngeal consonants Ain /ʕ/ and Haa /ħ/ and were 72% and 76.9%, respectively. The recognition rates for the uvular consonants Ghain /ɣ/, Qaa /q/, and Khaa /x/ were 64%, 55%, and 86.7% respectively. Arabic phoneme classification using HMM is presented in [63]. Experiments were conducted on The Holy Quran (THQ) corpus, which has been prepared from THQ recitations and is segmented semi-manually at the phoneme level. Four systems were designed, namely monophone, left context biphone, right context biphone, and triphone, and correction rates of 76.04%, 93.01%, 93.59%, and 92.81%, respectively, were obtained.

Similar to those of other languages, Arabic ASR systems using deep learning have received considerable attention recently. The Arabic ASR system using the Kaldi toolkit for a large broadcast corpus called GALE, which contains 200 hours of recordings, is presented in [64]. This corpus contains recordings of broadcast reports and broadcast conversations. Different models were proposed, and the best WERs were obtained using DNN + MPE; they were 15.81%, 32.21%, and 26.95% on the broadcast reports, broadcast conversations, and broadcast reports with broadcast conversations, respectively.

III. METHODOLOGY

In this section, we discuss in detail the research methodology applied in this study, including speech corpora and evaluation metrics, data preparation, proposed systems, and training/testing phases. We start by describing in brief the speech corpora and evaluation metrics that were used in this study. The data preparation process in which raw waves are converted into images will be explained.

Then, a general overview of the proposed systems will be given. We will then explain the training process of YOLO and CenterNet. The adaptation of these detectors to the phoneme recognition task is also provided. Each of the two detectors was used in the DTS, IaTS, and IeTS. For the DTS and IaTS, we used two backbone networks with each of the two detectors. For IeTS, we used one backbone network for each of the two detectors. Finally, we describe the testing process as well as the post processing used to generate the sequence of phonemes from the detector's output. A general overview of the methodology steps is shown in Figure 1.

A. SPEECH CORPORA

To evaluate the performance of our proposed methods, three speech corpora are used in this investigation, two from English language and one from Arabic language. A brief description of each corpus is presented in this section.

1) TIMIT CORPUS

The TIMIT corpus [65], developed by Texas Instruments (TI), Massachusetts Institute of Technology (MIT), and the Stanford Research Institute (SRI), is a well-known and popular corpus; hence, we used it to investigate our proposed methods. TIMIT comprises the recordings of 630 speakers. Each speaker read ten sentences, and the total duration of all recording is about 5.4 h. TIMIT defined the corpus partition in the training and test sets as 462 speakers for training set, 24 speakers for core test set, and 168 speakers for complete test set. In our experiments, we followed the configuration of the TIMIT defined by Kaldi. The Kaldi configuration excluded the two dialect sentences SA1 and SA2 from all sets as suggested in most of the previous studies. We used 48 phones in the training and testing phases, and these phones were mapped to 39 phones during evaluation as suggested in [66].

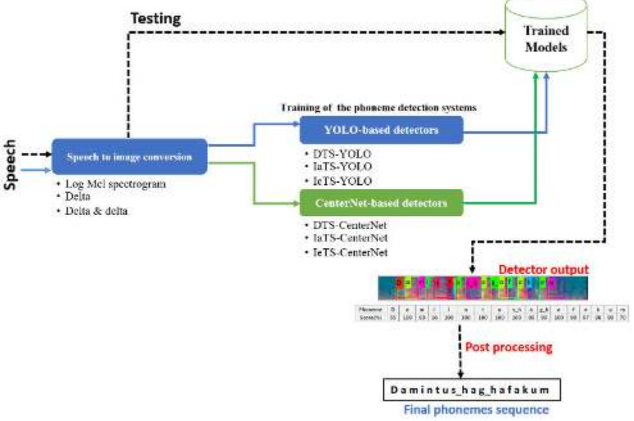


FIGURE 1. Overview of the proposed methodology.

2) LIBRISPEECH CORPUS

The LibriSpeech corpus [14] was used in our study for the Intra-language transfer learning task. LibriSpeech is a free corpus that contains the recordings of 1000 hours of read speech. In our experiments, we randomly selected the 15.7 h of recordings for training and 3.85 h for validation from 50 speakers from the *train-clean-100* subset. The selected set contains utterances from 26 males and 24 females. We used this corpus to overcome the scarcity of the data on the TIMIT corpus and to study the effect of transfer learning within the same language. To adapt the LibriSpeech corpus to fulfill the requirements of our research, we had to segment it to the phone level before using it. To do that, we used an open source force alignment tool called the Montreal forced aligner (MFA) [67] for the phone segmentation task. For the MFA to work, it needs the acoustic model of the language of the speech. MFA developers have provided an acoustic model for many languages. They used the entire LibriSpeech corpus to build the acoustic model of the English language. Hence, this was one of the reasons for using the LibriSpeech corpus in our study.

3) KSU ARABIC SPEECH CORPUS

The King Saud University Arabic speech corpus [68] is a rich Arabic corpus owing to the number of speakers of different genders, ages, and nationalities, as well as the number of different recording environments and texts. In terms of gender, the corpus contains the recordings of 269 male and 95 female speakers. In terms of nationalities, speakers are

TABLE 1. Arabic phonemes and number of occurrences of each phoneme.

Arabic phoneme	English symbol	# of samples		Arabic phoneme	English symbol	# of samples	
		train	test			train	test
الف مد	a2	540	341	ظ	~Z	57	38
ب	b	157	104	ع	~@	293	168
ت	t	194	118	غ	g_h	117	64
ث	t_h	134	89	فتحة	a	1904	1100
ج	j	30	32	ف	f	337	205
ح	~h	211	105	ق	q	68	41
خ	x	66	43	ك	k	158	79
د	d	70	48	كسرة	i	547	349
ذ	~z	76	65	ل	l	552	339
ر	r	250	155	م	m	537	306
ز	z	92	55	ن	n	585	306
س	s	346	195	هـ	h	377	203
ش	s_h	121	65	همزة	@	352	209
ص	S	128	73	و	w	143	75
ض	D	54	44	واو مد	u2	24	20
ضمة	u	270	171	ي	y	110	80
ط	T	19	9	ياء مد	i2	127	73
		Total number of phonemes					**14413**

classified to Saudis, non-Saudi Arabs, and non-Arabs. With regard to recording environments, the recording was done in three environments (office, soundproof room, and cafeteria). Each speaker uttered 16 lists that varied between words, sentences, paragraphs, and spontaneous speech. As we needed segmented data at the phoneme level, in this study, we used a subset from the KSU corpus that was segmented at the phoneme level [69]. The total phonemes in this subset was 14413 with a total duration of recording of about 0.47 h. This small data size encouraged us to study the effect of transfer learning between different languages. We used 34 Arabic phonemes listed in [69]. The Arabic phonemes and the corresponding symbols and number of samples in the train and test sets are represented in Table 1.

B. EVALUATION METRICS

We used two metrics to evaluate our proposed systems. The first one belongs to the object detection domain, and it called mean average precision (mAP) which is a significant metric to measure the accuracy of object detection models. The second metric is phone error rate (PER) which belongs to the speech processing domain to evaluate the automatic speech recognition systems.

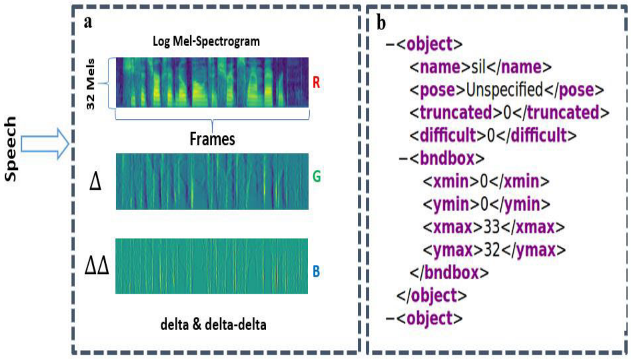
1) AVERAGE PRECISION (AP)

The AP is a universal metric that can be used to evaluate the performances of different systems, such as information retrieval systems [70], recommender systems [71], and object detection systems [72]. The AP is computed from the area under the precision-recall curve [73]. The precision and recall for each class are calculated using (1) and (2).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

To classify each predicted bounding box as a true positive, false positive, and false negative, we need to clarify an important term called the intersection over union (IoU), which is also called a Jaccard index. The IoU is the ratio between the area of intersection and the area of union of a ground truth bounding box and a predicted bounding box [74]. The algorithm used to calculate TP and FP using the IoU threshold is presented in [10]. Then, the AP for each class is computed as the average of the precision values over the interval of recall between "0" and "1" [75]. Then, the mAP is computed by the average of the AP over all the classes. In this study, we used the mAP_{50} term that means the average of the AP, which is calculated at an IoU threshold of 0.5.



```
-<object>
   <name>sil</name>
   <pose>Unspecified</pose>
   <truncated>0</truncated>
   <difficult>0</difficult>
  -<bndbox>
     <xmin>0</xmin>
     <ymin>0</ymin>
     <xmax>33</xmax>
     <ymax>32</ymax>
   </bndbox>
 </object>
-<object>
```

FIGURE 2. Data preparation: a) Speech to image conversion; b) Annotation file.

2) PHONE ERROR RATE (PER)

The word error rate is a popular metric in the speech processing field. For phone-level recognition systems, the word error rate is called a PER [76]. In our experiments, we used the HResults analysis tool from the HTK toolkit [77] to calculate PER, which is computed by (3).

$$PER = 100 - \left[\frac{H - I}{N} \times 100\%\right] \quad (3)$$

where Nrepresents the number of reference labels, H is the total number of correct labels, and I represents the number of insertions. Moreover, we used the correct rate metric for each phoneme p, which is calculated by (4) [63].

$$Correct\ rate(\%c) = \frac{H_p}{N_p - D_p} \times 100\% \quad (4)$$

where $H = N - S - D$, S represents the number of substitutions, and D is a number of deletions.

C. DATA PREPARATION

The purpose of this phase is to convert the raw wave utterances to images by using a spectrogram. In our work, phonemes are considered objects within bounding boxes, and these bounding boxes are given the corresponding annotation. While in normal speech processing, phonemes are defined by their start and end times, and this interval is given the corresponding annotation. This phase consists of two steps: in the first step, a 3-channel image is generated from the speech signal. Next, the annotation files that contain the phonemes and their respective bounding boxes are created as shown in Figure 2-a and Figure 2-b.

1) SPEECH TO IMAGE CONVERSION

Usually, the input of object detectors are images containing one or more objects with their respective bounding boxes. To use deep object detectors in the field of speech processing, we need to represent speech signals as images and represent the phonemes as objects within the images with their bounding boxes. The time-frequency signal representations, such as spectrograms, are frequently used for speech processing tasks, as described in [78]–[83]. In this work, we will generate an image with three channels that are the log Mel-spectrogram and its derivatives (delta and delta-delta), as in [84].

We used Librosa [85] to generate the log Mel-spectrograms from the speech signals. First, the speech signal is divided into overlapped frames using 256 samples (16 ms) for the frame length and 64 samples (4 ms) for the frame stride. The sampling rate of the signal was 16 kHz. We used the Hanning window [86]. A Fourier Transform of the windowed signal was calculated; then the power spectrum was calculated using (5) [86]:

$$PS(k) = \frac{1}{N^2}|F(k)|^2, \quad k = 0, 1, \ldots, N-1 \tag{5}$$

The power spectrum was passed through a 32 bandpass filter so that it could be converted to the Mel scale to calculate the Mel-spectrogram. Then, it was converted to the log scale (decibel unit). The first and second derivatives were calculated and appended to create a three-channel image that was visually displayed as an RGB image. Each channel was normalized to the range (0,1). The final output was an image with dimension (*number of Mels (height)*× *number of frames (width)*×*3 (channels)*), as shown in Figure2-a. For the TIMIT corpus, the maximum width size of the generated spectrograms was 1939, and the second one had 1885 frames. Some detectors require a size that is divisible by 32. Hence, by zero padding, we padded all channels of images to 1888, which is the closest number to the average of the two highest numbers that are divisible by 32. Detectors will resize all images that have a size greater than 1888. For the Arabic corpus, we observed that more than 98% of utterances have a width size less than 512, so we padded all channels of images to 512.

2) ANNOTATION FILE

In our experiments, we followed the Pascal VOC annotation format [29, 30]. Once the RGB images are created from the speech signals, the annotation files are extracted by creating the bounding boxes of each phoneme. The start and end times of each phoneme are provided by the phone transcript file, which is generally supplied with the speech corpora. The detailed steps of the technique used to create the images and annotate them is shown in Figure 3. We found the start and end frames of the phone from the phone transcript files, the indices of the start and end frames were *xmin* and *xmax* of the bounding box, respectively. We ignore any phoneme if its width is less than or equal one frame. Given that all phones have the same Mel-spectrogram height, we used ''0'' and number of Mels for *ymin* and *ymax* of the bounding box coordinates, respectively. Furthermore, the class label of each phoneme was occupied inside the annotation files as shown in Figure2-b, where the bounding box was for a silent phoneme.

The output of this phase is a database that contains ''images and annotation files,'' which is similar to the Pascal VOC and COCO databases.

D. THE PROPOSED SYSTEMS

We propose three systems, namely DTS, IaTS, and IeTS, to answer the research questions of this study. In this section, we give a general overview of the proposed systems.

1) DOMAIN TRANSFER LEARNING SYSTEM (DTS)

In DTS, we used ImageNet pre-trained weights as the initial weights for the backbone networks. This means that transfer learning from the image processing task to the speech processing task is presented as shown in Figure 4. In the training phase, the weights of the backbone networks are fine-tuned by the speech data of the TIMIT corpus in the form of images of three channels for phoneme recognition. While the weights of the detection layers are initialized randomly. This system allows us to evaluate the impact of transfer learning between the image and speech domains.

2) INTRA-LANGUAGE TRANSFER LEARNING SYSTEM (IaTS)

To overcome the issue of lack of data on the TIMIT corpus, we proposed a second system that is called IaTS. We investigated fine-tuning of the backbone networks using a corpus that is much larger than TIMIT. We chose a subset of LibriSpeech that is three times larger than TIMIT as explained in speech corpora section. First, as in the previous system, the weights of the backbone network were initialized using ImageNet weights. Then, LibriSpeech was used to train the backbone network and detection layers. The input was speech data from the LibriSpeech corpus as 3-channel images, and the output was the sequence of phonemes. Next, the TIMIT data were used to fine-tune the system for phoneme recognition, where the input were speech data from the TIMIT corpus as 3-channel images, and the output was the sequence of TIMIT phonemes. The pipeline of this system is shown in Figure 5. This system allows us to demonstrate the performance of intra language transfer learning between two corpora.

3) INTER-LANGUAGE TRANSFER LEARNING SYSTEM (IeTS)

Finally, the third proposed system in this study is IeTS. The Arabic speech dataset is very small; hence, it cannot be used to train the system (backbone and detection layers) from scratch. To address this, we used the weights of DTS, which was initialized by ImageNet and fine-tuned by TIMIT. We selected TIMIT as a source language in this system because it is a well-known corpus with published results and verified time labelling at the phoneme level.

```
Input: Speech corpus
  ▪ for each wave file in speech corpus do
  ▪ Generate log Mel-spectrogram, delta and delta-delta.
  ▪ Apply Normalization.
  ▪ Find the time in seconds for all frames frames_time[], using librosa.frames_to_time.
  ▪ for each line in associate phone transcript file do
      • find the minimum and maximum time of current phoneme, t_min, t_max.
      • Find index of start frame f_min by $\underset{i \in frames\_time[]}{\operatorname{argmin}} |frames\_times[i] - t\_min|$
      • Find index of end frame f_max by $\underset{i \in frames\_time[]}{\operatorname{argmin}} |frames\_times[i] - t\_max|$

      • // Compute object coordinates:
      • width = f_max - f_min, height = height of log Mel-spectrogram
      • Xmin = f_min, Xmax = f_min + width
      • Ymin = 0, Ymax = height
      • End if
  ▪ End for
  ▪ End for
Output: Database similar to PASCAL VOC
```

FIGURE 3. Data preparation steps.

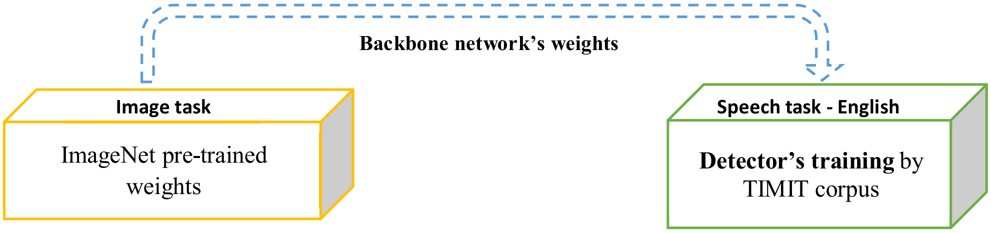

Backbone network's weights
Image task
ImageNet pre-trained weights
Speech task - English
Detector's training by TIMIT corpus

FIGURE 4. Pipeline of DTS.

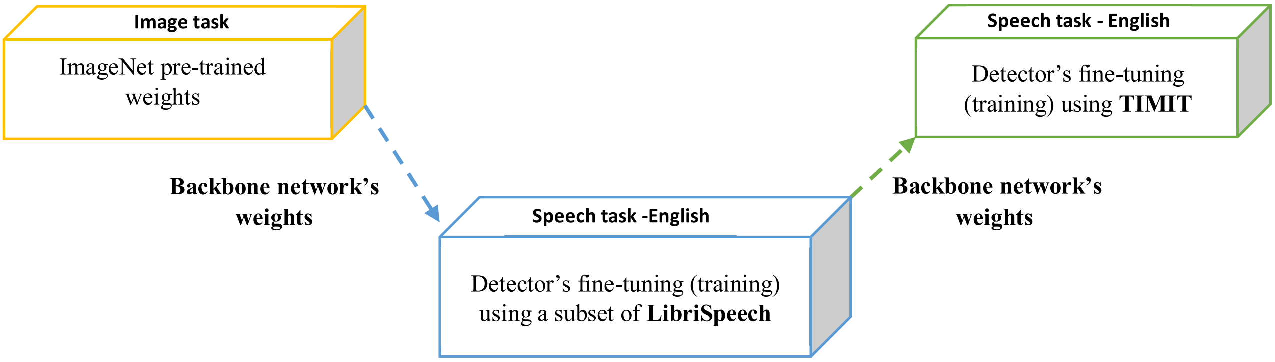


Image task
ImageNet pre-trained weights
Backbone network's weights
Speech task -English
Detector's fine-tuning (training) using a subset of LibriSpeech
Backbone network's weights
Speech task - English
Detector's fine-tuning (training) using TIMIT

FIGURE 5. Pipeline of IaTS.

We started from the network weights of DTS. Then, the system (backbone network and detection layers) was fine-tuned using the data from Arabic speech for input and output. We compared the performances of the first two systems with published results on TIMIT. This cannot be done for the IeTS system; therefore, we consider as a baseline a detector system with backbone networks initialized by ImageNet. The flow diagram of this system is shown in Figure 6.

For each of the proposed three systems, we investigated the use of state-of-the-art object detection techniques, namely YOLO and CenterNet, with different backbone networks. In the following sections, the training and testing processes of all the investigated proposed systems will be discussed.

E. TRAINING YOLO-BASED SYSTEMS

In this section, the adaptation of the YOLO object detector to phoneme sequence recognition is presented. The three proposed systems that use the YOLO detector are DTS-YOLO, IaTS-YOLO, and IeTS-YOLO, as shown in Figure 7. We investigated the use of two models of the

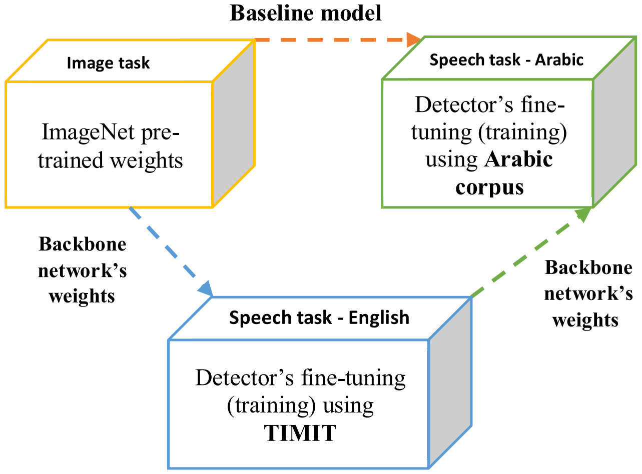

Baseline model
Image task
ImageNet pre-trained weights
Speech task - Arabic
Detector's fine-tuning (training) using **Arabic corpus**
Backbone network's weights
Backbone network's weights
Speech task - English
Detector's fine-tuning (training) using **TIMIT**

FIGURE 6. Pipeline of IeTS and its baseline model.

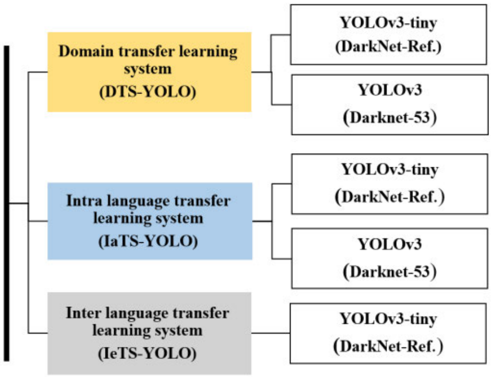

Domain transfer learning system (DTS-YOLO)
Intra language transfer learning system (IaTS-YOLO)
Inter language transfer learning system (IeTS-YOLO)
YOLOv3-tiny (DarkNet-Ref.)
YOLOv3 (Darknet-53)
YOLOv3-tiny (DarkNet-Ref.)
YOLOv3 (Darknet-53)
YOLOv3-tiny (DarkNet-Ref.)

FIGURE 7. YOLO detector of the three proposed systems.

YOLO detector, which are YOLOv3 and YOLOv3-tiny (with three scales). We used DarkNet-53 as the backbone network for YOLOv3, and DarkNet-reference as the backbone network for YOLOv3-tiny. For the last system (IeTS), we investigated the use of one model, that is, YOLOv3-tiny using a DarkNet-reference backbone. Figure 7 shows the proposed systems and the proposed models of each system.

1) BACKGROUND OF YOLO AND BACKBONE NETWORKS

The DarkNet framework is an open source framework developed by Josep Redmon to train YOLO detectors and perform other tasks. Darkent is fast, designed to support GPUs and CPUs, easy to install, and written in C and CUDA [87]. A variant of the DarkNet that was actively developed by AlexeyAB's GitHub fork [88] was used to train YOLO detectors.

As already mentioned, YOLO is a real-time one-stage object detector. Different versions of YOLO with different backbone networks exist. The first version was presented in 2015 and is called YOLOv1[8]. A faster version called Fast YOLO was proposed and contained a very small backbone network [8]. Chronologically, the YOLO team introduced a new version called YOLOv2 [23] by making some improvements on the previous model. These improvements were done by increasing the input resolution, using the batch normalization for each convolutional layer, and using the Anchor boxes. Moreover, they proposed YOLO9000 version, which is a real-time object detection technique for more than 9000 categories. More details about the improvements are presented in [23]. Recently, the YOLO team announced a new model called YOLOv3 as presented in [24] and made more enhancements to increase the accuracy and maintain the detection performance in real-time as well. They also proposed a deeper and more accurate backbone network called **DarkNet-53**, which has 53 convolutional layers. Compared to the residual networks, the authors stated that "Darknet-53 is better than ResNet-101 and 1.5× faster. Darknet-53 has a similar performance to ResNet-152 and is 2× faster". Furthermore, they made the prediction through three different scales to improve the performance of the detection. YOLOv3 supports multi-label classification by replacing the softmax by an independent logistic classifiers and using binary cross entropy loss for class predictions. YOLO is optimized in an end-to-end manner, and the loss consists of three parts as presented in [8] and explained in [89]. These parts are classification loss (CL), localization loss (LL), and objectness loss (OL). Hence, the YOLO loss function is described as a sum of all the three parts as tabulated in Table 2.

TABLE 2. YOLO loss function [8], [89].

$\mathbf{CL} = \sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{c\in classes}(p_i(c) - \hat{p}_i(c))^2$ **Where** $\mathbb{I}_i^{obj} = 1$, if object appears in cell i **and** S^2 denotes the number of grid cells
$\mathbf{LL} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\widehat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\widehat{h}_i})^2]$ **Where** (x, y, w, h) represent the center coordinates, width, and height of the bounding box. **B** is a number of predicted bounding boxes in each grid cell.
$\mathbf{OL} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2$ **Where** C represents the confidence score of bounding box.
YOLO loss = CL + LL + OL

2) YOLO TRAINING FOR DTS (DTS-YOLO)

We started by training the YOLOv3-tiny model for TIMIT phoneme sequence recognition. YOLOv3-tiny uses a small backbone network that is called a DarkNet-Reference, which is pre-trained by ImageNet. DarkNet-Reference is a CNN with fewer parameters than AlexNet, and on a par with AlexNet in terms of performance [90]. It has 13 consecutive convolutional and pooling layers, and achieved 61.1 and 83.0 for Top-1 and Top-5 accuracy on ImageNet, respectively.

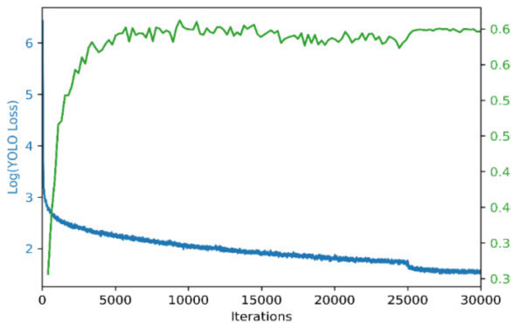


FIGURE 8. Training loss and mAP of YOLOv3-tiny for DTS-YOLO [ImageNet -> TIMIT].

We followed the same training strategy in the original paper of YOLO [8], where, as stated by the authors, extensive data augmentation was used to avoid overfitting. In our case, we used the same proposed data augmentation process, but the color, rotation, and flip augmentation functions were disabled because, after a lot of experiments the detection accuracy was decreasing rapidly with these augmentations. This was probably because our pixels are not real color codes; rather, they are a concatenation of three arrays that are represented as an RGB image. Furthermore, the rotation and flip augmentation functions changed the semantic meaning of the spectrogram unlike computer vision which is a semantic-preservation deformation [91].

We used the following training parameters: 0.001 for the learning rate, 0.9 for momentum, 0.0005 for decay, 64 for batch size, and 30,000 for the number of iterations. For the last 15% iterations, the learning rate was reduced by 10×. The loss curve of the TIMIT training set and the mAP of the TIMIT validation set are shown in Figure 8. We achieved an mAP of 0.65 at IoU = 0.5 after 30,000 iterations as shown on the left y-axis.

Furthermore, we plotted the log of the YOLO loss on the right y-axis. The x-axis represents the number of iterations. We also investigated the DTS-YOLO using the YOLOv3 model with DarkNet-53 for a backbone network. The training parameters were similar to those used to train YOLOv3-tiny.

We obtained the training loss curve and mAP performance shown in Figure 9. This model outperformed the previous tiny model and achieved nearly 0.75 in terms of mAP for the TIMIT validation set with a 15.3% improvement percentage.

3) YOLO TRAINING FOR IATS (IaTS-YOLO)

We started by training the YOLOv3 and YOLOv3-tiny models using the LibriSpeech corpus with training parameters similar to those of DTS-YOLO for 20,000 iterations. Then, we fine-tuned the models using the TIMIT corpus for another 10,000 iterations. Figure 10 shows the training loss and mAP for the YOLOv3-tiny. From the figure, we can see the mAP increased from 0.65 for DTS-YOLO to around 0.70 for IaTS-YOLO after only 10,000 iterations. This enhancement is due to adding the LibriSpeech pre-trained weights to the TIMIT phoneme sequence recognition task.

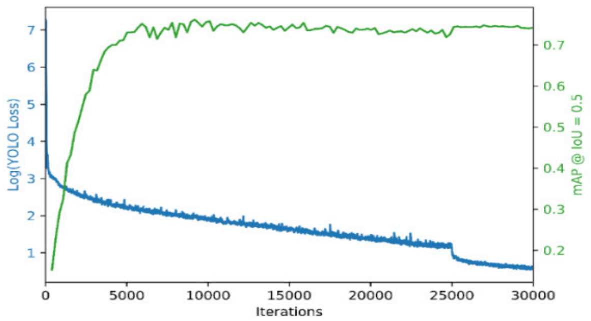


FIGURE 9. Training loss and mAP of YOLOv3 for DTS-YOLO [ImageNet -> TIMIT].

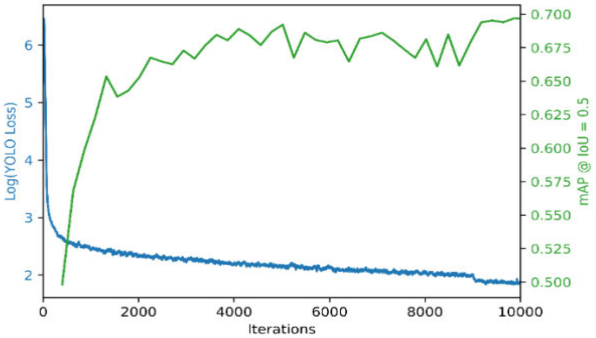


FIGURE 10. Training loss and mAP for YOLOv3-tiny, for IaST-YOLO [ImageNet -> LibriSpeech -> TIMIT].

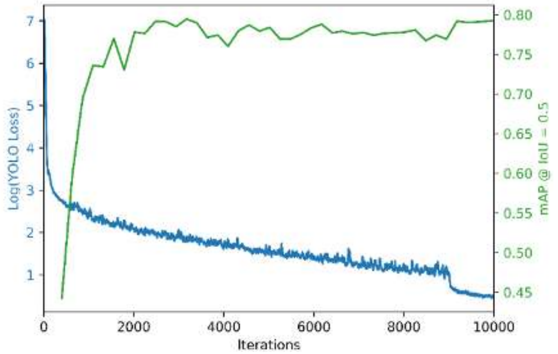


FIGURE 11. Training loss and mAP of YOLOv3 for IaTS-YOLO [ImageNet -> LibriSpeech -> TIMIT].

We repeated the experiment using YOLOv3 with the same training parameters and same number of iterations for LibriSpeech and TIMIT. The training loss and mAP curve are presented in Figure 11. The mAP of this model starts at 0.44 and reaches approximately 0.80, which is the best value achieved. This performance is better than that of the same model of a previous system (DTS-YOLO), where mAP started at 0.15 and reached 0.75 after 30,000 iterations as shown in Figure 9.

This performance indicates the benefit of using transfer learning between LibriSpeech and TIMIT corpora.

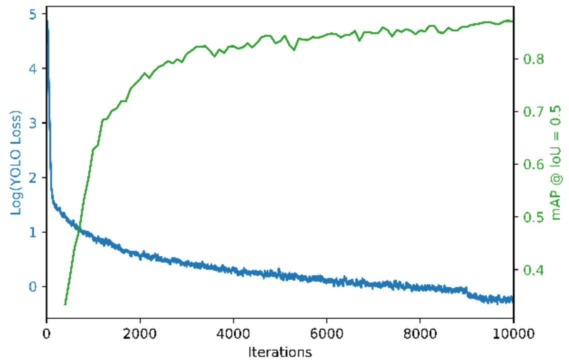


FIGURE 12. Training loss and mAP for YOLOv3-tiny for IeTS-YOLO [ImageNet -> TIMIT -> ARABIC].

4) YOLO TRAINING FOR IeTS (IeTS-YOLO)

IeTS-YOLO aims to study the effect of transfer learning from English to Arabic. Our Arabic corpus is small, and its classes are fewer than those of the TIMIT corpus as we presented in section III.A.3. Therefore, we investigated the use of only the tiny model of YOLO from DTS-YOLO for Arabic phoneme sequence recognition. We transferred the weight of the layers of the backbone network for DTS-YOLO to this system and we fine-tuned it using the small Arabic corpus. We used the training parameters of IaTS-YOLO. The curves of training loss and mAP are shown in Figure 12. We can clearly see that we achieved a high mAP of 0.87 after only 10,000 iterations.

F. TRAINING CENTERNET-BASED SYSTEMS

In addition to YOLO, we propose using another cutting edge and recently published object detector called CenterNet, which depends on the key points estimation strategy. Hence, in this phase, we discuss applying the CenterNet detector to the English and Arabic phoneme sequence recognition task using the three proposed systems, as for YOLO. We start this section by giving a brief background of the CenterNet detector. Then, the training process of the three proposed systems is presented in Figure 13. We investigated use of the two backbone networks of CenterNet, namely ResNet-18 and DLA-34.

1) BACKGROUND

Similar to YOLO, CenterNet is a one-stage detector that depends on keypoints estimation networks. It does not need to use prior anchors, as in the case of YOLOv3, and it does not require post processing after the detection process. CenterNet is presented in [9]. The authors proved that their detectors were faster, simpler, more accurate than bounding box based detectors, and could run in real time. We therefore considered it in our research. CenterNet starts to detect the object from the object center point and regresses the object size. The middle frames of phonemes are important in speech processing systems [92]. From the acoustic point of view, middle frames are more stable than other frames [93]. As CenterNet starts from the object center, it was an excellent candidate to use for phoneme recognition. The authors conducted experiments with four different backbone architectures, ResNet-18, ResNet-101, DLA-34, and Hourglass-104, which are different from the backbones used with YOLO.

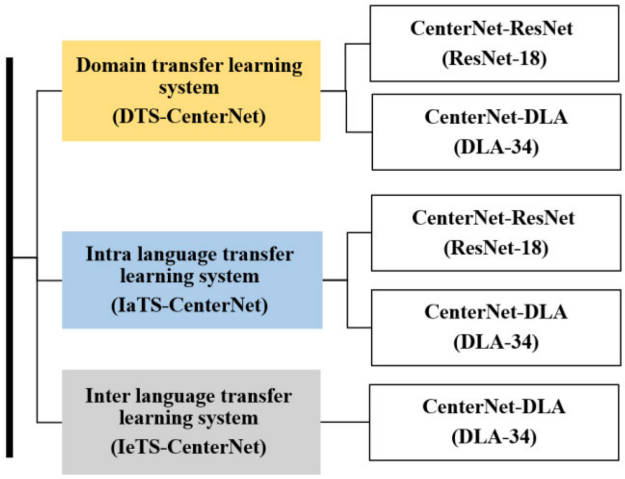

Domain transfer learning system (DTS-CenterNet)
CenterNet-ResNet (ResNet-18)
CenterNet-DLA (DLA-34)
Intra language transfer learning system (IaTS-CenterNet)
CenterNet-ResNet (ResNet-18)
CenterNet-DLA (DLA-34)
Inter language transfer learning system (IeTS-CenterNet)
CenterNet-DLA (DLA-34)

FIGURE 13. CenterNet detector for three proposed systems.

TABLE 3. Centernet loss function [9].

$L_{keypoint} = \frac{-1}{N}\sum_{xyc}\begin{cases}(1-\hat{Y}_{xyc})^{\alpha}\log(\hat{Y}_{xyc}) \\ \quad , \; if\; Y_{xyc}=1 \\ (1-Y_{xyc})^{\beta}\;(\hat{Y}_{xyc})^{\alpha}\log(1-\hat{Y}_{xyc}) \\ \quad , \quad otherwise\end{cases}$ **Where** N is the number of keypoints, α *and* β are the focal loss parameters.
$L_{offset} = \frac{1}{N}\sum_{P}\left\lvert\hat{O}_{\tilde{p}} - (\frac{p}{R} - \tilde{p})\right\rvert$ **Where** $\hat{O}$ is the predicted local offset of the keypoints location $\tilde{p}$
$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\lvert\hat{S}_{p_k} - s_k\rvert$ **Where** s_k is the object size for the center point p_k for object k
$\boldsymbol{CenterNet\_Loss} = L_{keypoint} + \gamma_{offset}L_{offset} + \gamma_{size}L_{size},$ Where γ_{offset} and γ_{size} are a constants for loss weighting. They used $\gamma_{offset} = 1$ and $\gamma_{size} = 0.1$

Our choice of backbone network depended on a tradeoff between accuracy and speed. Thus, we selected, for our phoneme sequence recognition task, two networks to examine, the ResNet-18 and the DLA-34.

Residual networks were proposed by He *et al.* [36] to solve the difficulties in training deeper neural networks. These networks won first place in many visual recognition challenges such as ILSVRC and COCO. Different architectures were proposed with different numbers of layers, such as 18 layers, 34 layers, 50 layers, 101 layers, and 152 layers. ResNet has also been used for pose estimation. For this purpose, up-convolutional layers have been added after the last convolution stage in ResNet as proposed in [94]. Furthermore, the creators of the CenterNet detector modified ResNet for pose estimation by adding a deformable convolutional layer [95] before each up-convolutional layers.

The other backbone network is called deep layer aggregation (DLA). DLA was proposed by Yu *et al.* [37] as an aggregation technique for fusing representations of network layers. They proposed two architectures of deep aggregation, iterative and hierarchal deep aggregation, and applied DLA networks for classification, dense prediction (e.g. semantic segmentation), etc. The creators of the CenterNet detector have used DLA for dense prediction with some adjustments, including adding extra skip connections and modifying the up-sampling layers in DLA by replacing convolutional layers with deformable convolutional layers.

The CenterNet detector depends on the keypoints estimation.

Therefore, backbone networks are used for generate a heatmap from the input image, and the center of each objects is found by the peaks of the heatmap. In our case, image with size ($\# frames \times \#mels \times 3$) is fed to the backbone network to produce a keypoints heatmap of size ($(\# frames/R) \times (\# mels/R) \times C$), where ($R = 4$) is the output stride mentioned in the original paper [9], and C is the number of classes (48 for TIMIT corpus and 34 for Arabic corpus). To train the keypoints estimation network, the ground truth keypoint p is transformed to the lower size of the keypoint heatmap $(\tilde{p}) = \lfloor p/R \rfloor$ using a Gaussian kernel as shown in [9]. In addition to the keypoints estimation, CenterNet predicts the local offset, which is important for overcoming the error caused by reducing the size of the input image to the size of a keypoints heatmap. Finally, the size of each object is predicted by regressing the object size from the center points. Hence, the CenterNet loss function is a sum of three components: keypoints loss $L_{keypoints}$, offset loss L_{offset}, and object size loss L_{size}. The equation of each loss part is presented in Table 3, which is taken from the original paper [9]. We used $\gamma_{offset} = 1$ and $\gamma_{size} = 0.1$ as suggested by the authors.

2) CENTERNET TRAINING FOR DTS (DTS-CENTERNET)

In the training phase, we followed the training process of the original paper on CenterNet [9] with some modification of the parameters. We attempted to use the same training parameters of the YOLO models as much as possible to make a fair comparison. We disabled the color augmentation as we had during YOLO training. The models were trained with a batch size of 16 and for 48 classes for English and 34 classes for Arabic. We used the following training parameters: number of epochs = 140, leaning rate = 0.001, and learning reduction (10x at 90 and 120 epochs). All losses were trained using the Adam optimizer. FIGURE 14 shows the CenterNet loss in the training phase for the two models: CenterNet-ResNet and CenterNet-DLA. We can clearly see that the loss convergence of the DLA performed better than ResNet.

3) CENTERNET TRAINING FOR IaTS (IaTS-CENTERNET)

In this system, we trained the CenterNet models using a LibriSpeech corpus for 50 epochs without learning rate reduction. Then, in the second phase, we fine-tuned the models using the TIMIT corpus for another 70 epochs with a 10× learning rate reduction at 45 and 60 epochs. We used training parameters similar to those used in the DTS-CenterNet experiment. The training losses of the models are shown in Figure 15. We can see that the CenterNet-DLA loss decreased from 2.07 to 0.28 after 70 epochs, showing better convergence compared to the DTS-CenterNet, which was presented in Figure 14.

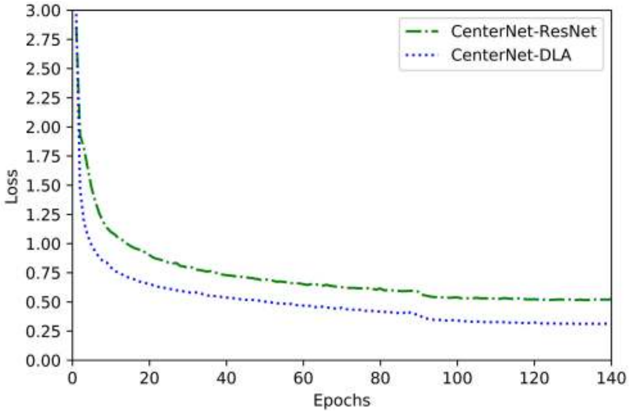


FIGURE 14. Training loss of CenterNet with backbones (ResNet and DLA), for DTS [ImageNet -> TIMIT].

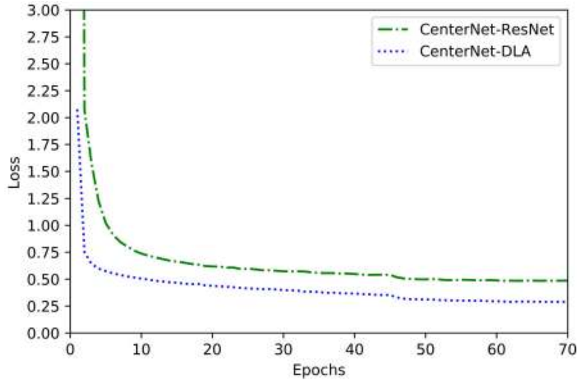


FIGURE 15. Training loss of CenterNet with backbones (ResNet and DLA), for IaTS [ImageNet -> LibriSpeech -> TIMIT].

4) CENTERNET TRAINING FOR IeTS (IeTS-CENTERNET)

In this model, we used the CenterNet detector for the Arabic phoneme recognition task. We investigated using only the DLA backbone network for the CenterNet detector. The pre-trained weights of the DLA were transferred from the DTS-CenterNet to this system to examine the effectiveness of transfer learning between English and Arabic languages. We used training parameters similar to those of the DTS-CenterNet. The training loss is shown in Figure 16.

G. TESTING THE PROPOSED SYSTEMS

The input of the testing phase is similar to the input of the training phase, which is a three-channel image of the speech signal for each utterance on the test set. The image of the entire utterance is fed to the trained detectors without any

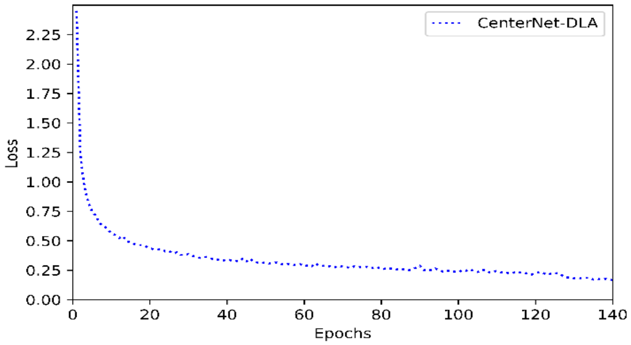


FIGURE 16. Training loss of CenterNet using DLA backbone, for IeTS [ImageNet -> TIMIT -> ARABIC].

time labelling as shown in Figure 17-a. Figure 17-b shows the output of the YOLOv3-IaTS system for sentence SX388 of speaker MTAS1 on the TIMIT core test set (after the mapping to 39 phonemes). The sequence of phonemes of the output detectors is written more clearly and shown in Figure 17-c. The corresponding output of the post processing is shown in Figure 17-d. The sequence of the phonemes after post-processing is written in a clearer form and shown in Figure 17-e. The reference phonemes with their bounding boxes are shown in Figure 17-f, and they are written in a clearer form in Figure 17-g.

As we can see from Figure 17-b, the output of the detector might have an overlap between bounding boxes of different classes. The overlap might be between bounding boxes of the same class or bounding boxes of different classes. This is acceptable in the object detection domain. In the object detection field, the researchers solve this by accepting an overlap between bounding boxes of different classes, while for an overlap between bounding boxes of the same class, they choose the bounding box with the highest confidence score using the NMS algorithm [96]. In our case, there should be no overlap between the bounding boxes of the phonemes, and phonemes should follow each other. Hence, we apply NMS and take the bounding box of the highest confidence score among all classes as shown in Figure 17-d. Then, the PER is calculated between the reference sequences of phonemes and the predicted output after post-processing, without using a language model. The calculated PER is 22% with the post processing step and is 28% without post processing.

IV. RESULTS AND DISCUSSION

First, we present the performance of the domain transfer learning system (DTS) for TIMIT phoneme sequence recognition. Then, the performance of intra-language transfer learning (IaST) between LibriSpeech and TIMIT for the TIMIT phoneme recognition task is presented. Finally, the performance of inter-language transfer learning (IeTS) between English and Arabic for Arabic phoneme sequence recognition task is presented.

TABLE 4. mAP and PER of DTS using TIMIT core test set.

System	Model	PER (%)	mAP$_{50}$(%)	Backbone
DTS-YOLO	YOLOv3-tiny	28.25	63.60	DarkNet-Ref.
	YOLOv3	20.2	73.25	DarkNet-53
DTS-CenterNet	CenterNet-ResNet	21.09	69.79	ResNet-18
	CenterNet-DLA	**19.06**	**73.41**	DLA-34

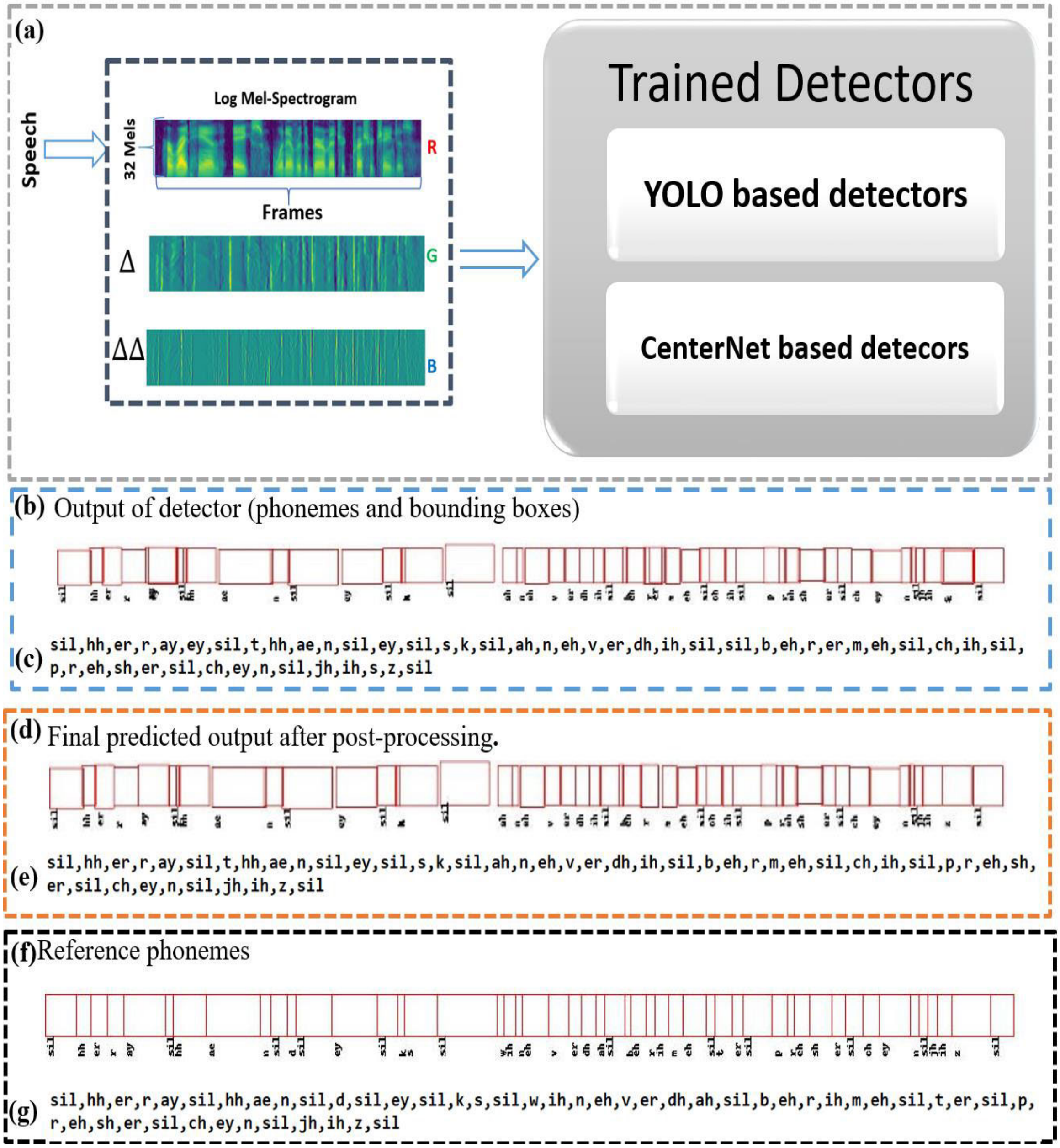
A. DOMAIN TRANSFER LEARNING SYSTEM (DTS)

We evaluated YOLO and CenterNet detectors for the phoneme sequence recognition task using our proposed DTS system. The performance of the system in this experiment is presented in Table 4. Four models were evaluated: two with YOLO detectors, which are presented in the first two rows, and two models with the CenterNet detectors are shown in the last two rows. All the models were evaluated using the TIMIT core test set. From Table 4, we can observe that YOLOv3 and CenterNet-DLA had almost the same values in terms of mAP. In terms of PER, CenterNet-DLA outperformed all other detectors by achieving a PER of 19.06%. The obtained PER results were 28.25%, 20.2%, 21.09%, and 19.06%, respectively, and the mAP results were 63.60, 73.25, 69.79, and 73.41, respectively, for YOLOv3-tiny, YOLOv3, CenterNet-ResNet, and CenterNet-DLA. Hence, we can observe that the mAP is inversely proportional to the PER. Moreover, we can observe that keypoint-based detectors (i.e., CenterNet) outperformed the bounding box-based detectors (i.e., YOLO) in the phoneme sequence recognition task.

B. INTRA-LANGUAGE TRANSFER LEARNING SYSTEM (IaTS)

The best (lowest) PER in the previous system was 19.06%, which is higher than those of the best state-of-the-art methods but comparable to other state-of-the-art methods. This was encouraging to us, so we investigated whether we could improve on this by using Intra-language transfer learning between different English corpora as in the IaTS. The performance of the IaTS is presented in Table 5. We observed that the PER significantly decreased for the DTS especially for the YOLOv3 and CenterNet-DLA models, which achieved PERs of 16.34% and 15.89%, respectively.

As in the DTS system, the CenterNet-DLA achieved the best performance with PER = 15.89% (16.6% improvement from the DTS). This improvement was attributed to the power of transfer learning between LibriSpeech and the TIMIT corpora. Because, the YOLO detector depends on predefined anchors, unlike the CenterNet detector, we achieved the highest mAP using the YOLOv3 detector, which suggests that bounding box-based detectors can outperform the keypoint-based detectors in terms of mAP. For our task,



(c) sil,hh,er,r,ay,ey,sil,t,hh,ae,n,sil,ey,sil,s,k,sil,ah,n,eh,v,er,dh,ih,sil,sil,b,eh,r,er,m,eh,sil,ch,ih,sil,
p,r,eh,sh,er,sil,ch,ey,n,sil,jh,ih,s,z,sil

(e) sil,hh,er,r,ay,sil,t,hh,ae,n,sil,ey,sil,s,k,sil,ah,n,eh,v,er,dh,ih,sil,b,eh,r,m,eh,sil,ch,ih,sil,p,r,eh,sh,
er,sil,ch,ey,n,sil,jh,ih,z,sil

(g) sil,hh,er,r,ay,sil,hh,ae,n,sil,d,sil,ey,sil,k,s,sil,w,ih,n,eh,v,er,dh,ah,sil,b,eh,r,ih,m,eh,sil,t,er,sil,p,
r,eh,sh,er,sil,ch,ey,n,sil,jh,ih,z,sil

FIGURE 17. Process of testing phase.

TABLE 5. mAP and PER of IaTS using TIMIT core test set.

System	Model	PER(%)	mAP_{50}(%)	Backbone
IaTS-YOLO	YOLOv3-tiny	25.57	68.01	DarkNet-Ref.
	YOLOv3	**16.34**	78.56	DarkNet-53
IaTS-CenterNet	CenterNet-ResNet	17.16	73.11	ResNet-18
	CenterNet-DLA	**15.89**	76.64	DLA-34

PER is more important than mAP; hence, CenterNet-based systems have better performances than YOLO-based systems. We also observed that PER is inversely proportional to the mAP but only for the same systems, while in the DTS system, it was inversely proportional in general, as shown on Table 5.

The confusion matrix of the 39 phonemes (classes) of the TIMIT corpus when the system with the best performance, CenterNet-DLA, is used is presented in Figure 18. We can clearly see that most of the phonemes were classified correctly with high confidence. Furthermore, we notice that the confused phonemes have the same articulatory features. For example, phoneme /m/ is confused with phoneme /n/, as both have the same nasal manner, both are anterior and voiced, and the place of articulation of phoneme /m/ is labial, while

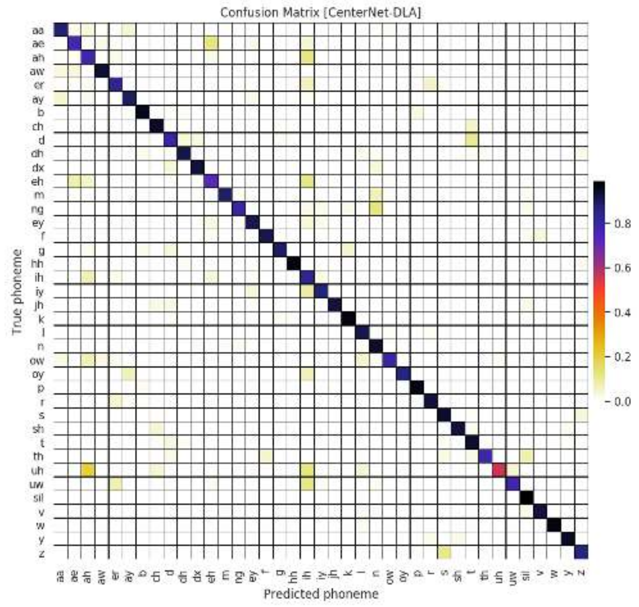


FIGURE 18. Confusion matrix of TIMIT phonemes using CenterNet-DLA model of IaTS.

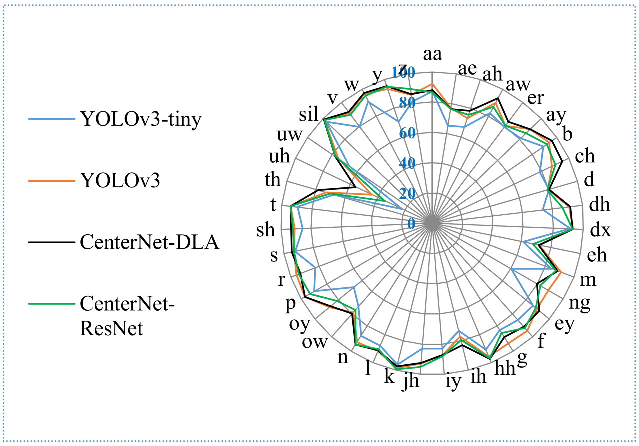


FIGURE 19. Percentage of correctness of each TIMIT phoneme using four proposed models in IaTS.

that of phoneme /n/ is a coronal. Furthermore, phoneme /z/ is confused with phoneme /s/, as both have the same place of articulation (coronal) and the same fricative classes [97]. Moreover, the confusions between vowels is more than the other phonemes. For example, phoneme /ih/ is confused with phonemes /ah/, /eh/, and /iy/. Furthermore, phoneme /uh/ is highly confused with phoneme /ah/.

The correct rate of each phoneme is calculated for all four detectors as shown in Figure 19. Most of phonemes have a correction rate greater than 80%. We can see that CenterNet-DLA outperforms all other detectors, and when it is used, more than 50% of the phonemes have a correction rate greater than 90%.

Using the CenterNet-DLA model, all the phonemes achieved a correct rate of more than 80%, except /uh/, /eh/, /ae/, /uw/, /th/, /ah/, and /ow/. All of these phonemes are vowels, except /th/. This observation may open a new direction for future work to examine different features instead of the Mel spectrogram to enhance the recognition of vowels.

TABLE 6. PER comparison between state-of-the-art and proposed IaTS.

Model	PER (%)	Year of publication
DNN-based multi-resolution spectrogram [48]	18.25	2018
Quaternion CNN [53]	19.64	2018
Hierarchical CNN [50]	16.5	2015
CNN [49]	20.36	2014
SincNet-Raw waveform [52]	17.2	2018
Attention-Based Model [1]	17.6	2015
LSTM [44]	**15.02**	2018
FFDNN [98]	18.0	2019
SPEECH-XLNET [99]	**13.3**	2019
Convolutional SVM [51]	16.9	2019
PYTORCH-KALDI [56]	**13.8**	2019
Deep-residual-networks [100]	16.0	2020
Yolov3 (Our)	**16.34**	
CenterNet-DLA (Our)	**15.89**	

C. COMPARISON OF PROPOSED IaTS WITH STATE-OF-THE-ART METHODS

Experiments of TIMIT phone sequence recognition have been conducted and discussed in previous sections. Now, we compare the performance of our best proposed models using our proposed systems with state-of-the-art. Previous studies on TIMIT phoneme recognition can be grouped into two categories. The first category used audio files and their transcripts, at the training and testing phases, without using the time boundaries of the phones. While the second category used phone time boundaries at train and test phases. In our work, we tackled the problem differently, our models required time boundaries only at the training level, while at the testing phase, only the audio of the whole utterance is required, aiming to predict the phonemes and their time boundaries. Hence, our comparison is with the first category, as presented in Table 6. We want to mention that most of the first category methods need a language model to work while our proposed methods work without a language model.

From Table 6, we notice that our CenterNet-DLA model performs better than most (75%) of state-of-the-art methods except three methods. We discuss here the difference between our models and the best three approaches. Firstly, the DNN model proposed in the paper [44] achieved 15.02% PER using LSTM. This was done using a bi-gram language/phone model for decoding. Pytorch-kaldi toolkit [56] reached 13.8% PER, which was the best published result for that time in TIMIT phone recognition task. This result is realized by a combination of different networks such as Li-GRU and MLP and combination of different features such as MFCC, FBANK, and fMLLR. For decoding, an n-gram language model was used. Lastly, the best result up to date for TIMIT phone recognition was reported by Speech-XLNET [99]. Speech-XLNET achieved 13.3% PER on TIMIT core test set using self-attention network (SAN). Pre-training was conducted using three large unlabeled corpora namely LibriSpeech, TED-LIUM release2, and WSJ-si284 using four GPUs (Tesla M40). They used a bigram phone language model for decoding.

The main goal of our work is investigating the use of object detection techniques for phoneme recognition task and we found that it gave an excellent results and outperformed many state of the art systems. The systems that outperformed our proposed systems used many features that we did not use. For example, all of them used language model while in our case we did not use a language model. Some of them used combination of different speech features, while we used only Mel spectrogram and its derivatives. The one with the best performance used pre-training with a three large corpora.

Likewise, we observe that the proposed YOLOv3 model is on a par with most of stat-of-the-art models such as [50, 51, 100] and outperforms many of cutting edge models such as [1], [48], [52], [53].

D. INTER-LANGUAGE TRANSFER LEARNING SYSTEM (IeTS)

We used a subset from the KSU Speech corpus that researchers have published results on [69]. The authors used a subset of the KSU corpus that contains utterances from native and non-native speakers. We compared our results with the results of the best experiments in [69] (Experiment 5.b). For training, the authors used 32 native speakers (16 were time labeled) and 5 time-labeled nonnative speakers. For testing, they used 11 non-native speakers. In our experiment, we needed only the time-labelled speakers. For the native speakers, we found only the data for 15 of the 16 speakers they used. Hence, in our work, we used only 15 native speakers and 5 nonnative speakers for training (time labeled) and 11 nonnative speakers for testing (non-time labeled). Table 7 shows the performance of the proposed models with TIMIT pre-trained weights and the performance of the baselines of the ImageNet pre-trained weights as defined in section III.D.3.

From Table 7, we can see the our proposed systems achieved an excellent PER result compared to the traditional method [69]. Comparing our proposed systems with the baselines, the proposed YOLO based system has a slightly better performance (around 1%) than the YOLO-based baseline with regards to the PER and an mAP of around 2%. For the CenterNet-based system, there was not much of a difference between the results of the proposed system and the corresponding baseline. We cannot argue that there is no effect by applying transfer learning between dissimilar languages duo to the small size of test set of an Arabic speech corpus. The confusion matrix of 34 Arabic phonemes for the best model, which is a CenterNet-DLA, is presented in Figure 20.

TABLE 7. mAP and PER for KSU arabic speech corpus.

System	Model	PER	mAP_{50}	Pre-trained weights	Backbone
IeTS-YOLO	YOLOv3-tiny (baseline)	11.14	85	ImageNet	DarkNet-Ref
	YOLOv3-tiny	10.15	**87**	DTS-YOLO	DarkNet-Ref
IeTS-CenterNet	CenterNet-DLA (baseline)	7.66	86.07	ImageNet	DLA-34
	CenterNet-DLA	**7.58**	85.71	DTS-CenterNet	**DLA-34**
HMM	[69]	28.8	-	-	-

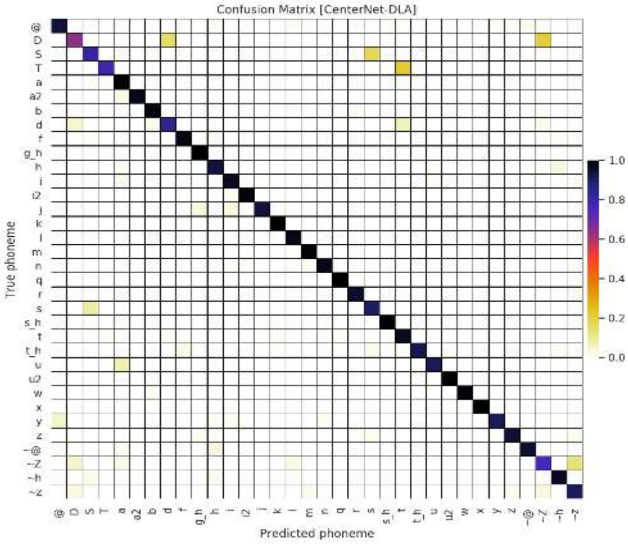


FIGURE 20. Confusion matrix of Arabic phoneme recognition using the CenterNet-DLA model of IeTS.

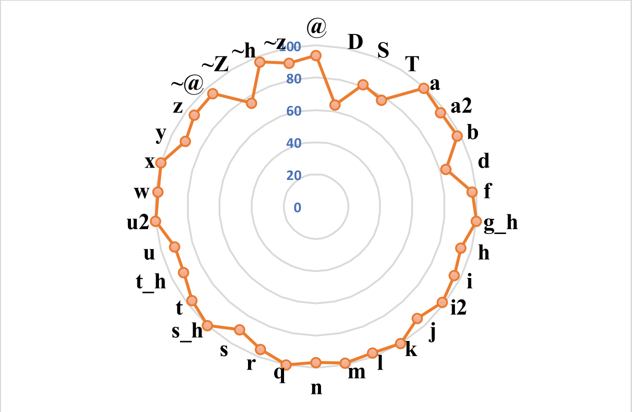


FIGURE 21. Correction rate of Arabic phonemes using the CenterNet-DLA model.

We notice that confusion occurred with phonemes that have similar articulatory features. For example, the phone /D/ is confused with phone /∼Z/ and the two phones are emphatic and voiced and very near in place of articulation. Furthermore, the phone /T/ is confused with phone /t/, as both are stops and have the same place of articulation (alveo-dental); furthermore, both are unvoiced, and while /T/ is emphatic, /t/ is non-emphatic. Phone /S/ is confused with phone /s/, as both are fricatives and have the same place of articulation (alveo-dental); furthermore, both are unvoiced, and while /S/ is emphatic, /s/ is non-emphatic [62].

The correct rate of each phoneme is calculated for the best model, CenterNet-DLA, and presented in Figure 21. All phonemes have a correction rate greater than 90%, except five phonemes, which are /D/, /∼Z/, /T/, /S/, and /d/. From the 34 phonemes, 20 phonemes (58.8%) achieved a correction rate greater than 95%. This excellent result shows the effectiveness of our third proposed IeTS system.

V. CONCLUSION

In this study, we proposed and investigated the use of object detection techniques for phoneme sequence recognition. We selected two state-of-the art real-time detectors, YOLO and CenterNet, and adapted them to phone sequence recognition in the English and Arabic languages. To overcome the scarcity of training data, three systems (DTS, IaTS, and IeTS) were proposed using different transfer learning scenarios. Two evaluation metrics, one from the speech recognition area and another from the object detection domain, were used to evaluate the performance of the proposed systems. For English phoneme recognition, we conducted the experiments on the TIMIT corpus. The results showed the effectiveness of using object detection techniques for phoneme sequence recognition. A comparison of 12 state-of-the-art methods for TIMIT phone recognition was presented, and our method outperformed most of those state-of-the-art techniques. For Arabic phoneme recognition, a subset of the KSU speech corpus was used. We compared our results with the available published results. Two models were examined and they outperformed the published results. From the results of the proposed systems, we can conclude that object detection techniques can be applied successfully to speech recognition tasks, especially to phoneme sequence recognition. Furthermore, transfer learning between corpora within the same language or between corpora from different languages improved the results.

In future work, we will investigate the effect of adding a language model to the proposed system. We anticipate this enhancement will produce better results and the resultant system may outperform state-of-the art methods with the best published results.

REFERENCES

[1] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[2] M. Alsulaiman, A. Mahmood, and G. Muhammad, “Speaker recognition based on arabic phonemes,” *Speech Commun.*, vol. 86, pp. 42–51, Feb. 2017.

[3] F. Nazir, M. N. Majeed, M. A. Ghazanfar, and M. Maqsood, “Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for arabic phonemes,” *IEEE Access*, vol. 7, pp. 52589–52608, 2019.

[4] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, “Sound classification using convolutional neural network and tensor deep stacking network,” *IEEE Access*, vol. 7, pp. 7717–7727, 2019.

[5] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” *IEEE Access*, vol. 7, pp. 128837–128868, 2019.

[6] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” 2019, *arXiv:1905.05055*. [Online]. Available: http://arxiv.org/abs/1905.05055

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[9] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” 2019, *arXiv:1904.07850*. [Online]. Available: http://arxiv.org/abs/1904.07850

[10] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, “Deep learning for generic object detection: A survey,” 2018, *arXiv:1809.02165*. [Online]. Available: http://arxiv.org/abs/1809.02165

[11] D. Han, Q. Liu, and W. Fan, “A new image classification method using CNN transfer learning and Web data augmentation,” *Expert Syst. Appl.*, vol. 95, pp. 43–56, Apr. 2018.

[12] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, “Semantic segmentation with transfer learning for off-road autonomous driving,” *Sensors*, vol. 19, no. 11, p. 2577, 2019.

[13] I. Athanasiadis, P. Mousouliotis, and L. Petrou, “A framework of transfer learning in object detection for embedded systems,” 2018, *arXiv:1811.04863*. [Online]. Available: http://arxiv.org/abs/1811.04863

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: An ASR corpus based on public domain audio books,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[15] Y. Segal, T. Sylvia Fuchs, and J. Keshet, “SpeechYOLO: Detection and localization of speech objects,” 2019, *arXiv:1904.07704*. [Online]. Available: http://arxiv.org/abs/1904.07704

[16] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Dec. 2001, p. 1.

[17] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[18] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[19] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[20] 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. 9–919.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[23] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[24] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, and C.-Y. Fu, “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2980–2988.

[27] H. Law and J. Deng, “CornerNet: Detecting objects as paired keypoints,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 50–734.

[28] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint triplets for object detection,” 2019, *arXiv:1904.08189*. [Online]. Available: http://arxiv.org/abs/1904.08189

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[30] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, and D. Ramanan, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[33] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, and A. Kuznetsova. (2017). *OpenImages: A Public Dataset for Large-Scale Multi-Label and Multi-Class Image Classification*. [Online]. Available: https://github.com/openimages

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[37] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2403–2412.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[39] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.

[40] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NASA STI/Recon 93, 1993.

[41] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: An automatic speech recognition dedicated corpus," in *Proc. LREC*, May 2012, pp. 125–129.

[42] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Workshop Speech Natural Lang. (HLT)*, 1992, pp. 357–362.

[43] M. Korvas, O. Plátek, O. Dušek, L. ilka, and F. Jur í ek, "Free English and Czech telephone speech corpus shared under the CC-BY-SA 3.0 license," in *Proc. 9th Int. Conf. Lang. Resour. Eval. (LREC)*, 2014, pp. 4423–4428.

[44] J. Michálek and J. Van k, "A survey of recent DNN architectures on the TIMIT phone recognition task," in *Proc. Int. Conf. Text Speech Dialogue*, 2018, pp. 436–444.

[45] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, and N. Jaitly, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, to be published.

[46] A.-R. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Proc. Workshop Deep Learn. Speech Recognit. Rel. Appl. (NIPS)*, 2009, p. 39.

[47] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[48] D. T. Toledano, M. P. Fernández-Gallego, and A. Lozano-Diez, "Multiresolution speech analysis for automatic speech recognition using deep neural networks: Experiments on TIMIT," *PLoS ONE*, vol. 13, no. 10, 2018, Art. no. e0205355.

[49] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech Language Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2015.

[50] L. Tóth, "Phone recognition with hierarchical convolutional deep maxout networks," *EURASIP J. Audio, Speech, Music Process.*, vol. 2015, no. 1, p. 25, Dec. 2015.

[51] V. Passricha and R. K. Aggarwal, "Convolutional support vector machines for speech recognition," *Int. J. Speech Technol.*, vol. 22, no. 3, pp. 601–609, Sep. 2019.

[52] M. Ravanelli and Y. Bengio, "Interpretable convolutional filters with SincNet," 2018, *arXiv:1811.09725*. [Online]. Available: http://arxiv.org/abs/1811.09725

[53] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linarès, R. De Mori, and Y. Bengio, "Quaternion convolutional neural networks for End-to-End automatic speech recognition," 2018, *arXiv:1806.07789*. [Online]. Available: http://arxiv.org/abs/1806.07789

[54] Y. Zhao, X. Jin, and X. Hu, "Recurrent convolutional neural network for speech processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 5300–5304.

[55] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 2, pp. 92–102, Apr. 2018.

[56] M. Ravanelli, T. Parcollet, and Y. Bengio, "The PyTorch-Kaldi speech recognition toolkit," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6465–6469.

[57] S. M. Abdou and A. M. Moussa, "Arabic speech recognition: Challenges and state of the art," *Comput. Linguistics Speech Image Process. Arabic Lang.*, vol. 4, p. 1, Mar. 2018.

[58] M. Alghamdi, M. Elshafei, and H. Al-Muhtaseb, "Arabic broadcast news transcription system," *Int. J. Speech Technol.*, vol. 10, no. 4, pp. 183–195, Dec. 2007.

[59] M. Ali, M. Elshafei, M. Al-Ghamdi, and H. Al-Muhtaseb, "Arabic phonetic dictionaries for speech recognition," *J. Inf. Technol. Res.*, vol. 2, no. 4, pp. 67–80, Oct. 2009.

[60] M. Y. El Amrani, M. M. H. Rahman, M. R. Wahiddin, and A. Shah, "Building CMU Sphinx language model for the Holy Quran using simplified arabic phonemes," *Egyptian Informat. J.*, vol. 17, no. 3, pp. 305–314, Nov. 2016.

[61] G. Muhammad, M. Masud, A. Alelaiwi, M. A. Rahman, A. Karime, A. Alamri, and M. S. Hossain, "Spectro-temporal directional derivative based automatic speech recognition for a serious game scenario," *Multimedia Tools Appl.*, vol. 74, no. 14, pp. 5313–5327, Jul. 2015.

[62] Y. A. Alotaibi and G. Muhammad, "Study on pharyngeal and uvular consonants in foreign accented arabic for ASR," *Comput. Speech Lang.*, vol. 24, no. 2, pp. 219–231, Apr. 2010.

[63] Y. Alotaibi, A. Meftah, and S.-A. Selouani, "Classical arabic phoneme contextual analysis using HMM classifiers," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2016, pp. 11–15.

[64] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, "A complete KALDI recipe for building arabic speech recognition systems," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 525–529.

[65] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, *TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1*. Philadelphia, PA, USA: Linguistic Data Consortium, 1993.

[66] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 11, pp. 1641–1648, Nov. 1989.

[67] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable text-speech alignment using Kaldi," in *Proc. Interspeech*, Aug. 2017, pp. 498–502.

[68] M. Alsulaiman, G. Muhammad, M. A. Bencherif, A. Mahmood, and Z. Ali, "KSU rich Arabic speech database," *Information*, vol. 16, pp. 4231–4253, Jun. 2013.

[69] A. Al Hindi, M. Alsulaiman, G. Muhammad, and S. Al-Kahtani, "Automatic pronunciation error detection of nonnative arabic speech," in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2014, pp. 190–197.

[70] B. Mitra and N. Craswell, "An introduction to neural information retrieval," *Found. Trends Inf. Retr.*, vol. 13, no. 1, pp. 1–126, 2018.

[71] I. Portugal, P. Alencar, and D. Cowan, "The use of machine learning algorithms in recommender systems: A systematic review," *Expert Syst. Appl.*, vol. 97, pp. 205–227, May 2018.

[72] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[73] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, Jan. 2016.

[74] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[75] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[76] D. A. Moses, N. Mesgarani, M. K. Leonard, and E. F. Chang, "Neural speech recognition: Continuous phoneme decoding using spatiotemporal representations of human cortical activity," *J. Neural Eng.*, vol. 13, no. 5, Oct. 2016, Art. no. 056004.

[77] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, and X. Liu, *The HTK Book*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 2002, p. 175.

[78] G. Kovács, L. Tóth, D. Van Compernolle, and S. Ganapathy, "Increasing the robustness of CNN acoustic models using autoregressive moving average spectrogram features and channel dropout," *Pattern Recognit. Lett.*, vol. 100, pp. 44–50, Dec. 2017.

[79] A. Satt, S. Rozenberg, and R. Hoory, "Efficient emotion recognition from speech using deep learning on spectrograms," in *Proc. Interspeech*, Aug. 2017, pp. 1089–1093.

[80] A. Haque, M. Guo, and P. Verma, "Conditional end-to-end audio transforms," 2018, *arXiv:1804.00047*. [Online]. Available: http://arxiv.org/abs/1804.00047

[81] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment sound classification using a two-stream CNN based on decision-level fusion," *Sensors*, vol. 19, no. 7, p. 1733, 2019.

[82] Y. Niu, D. Zou, Y. Niu, Z. He, and H. Tan, "A breakthrough in speech emotion recognition using deep retinal convolution neural networks," 2017, *arXiv:1707.09917*. [Online]. Available: http://arxiv.org/abs/1707.09917

[83] A. V. Oppenheim, "Speech spectrograms using the fast Fourier transform," *IEEE Spectr.*, vol. 7, no. 8, pp. 57–62, Aug. 1970.

[84] S. Zhang, S. Zhang, T. Huang, and W. Gao, "Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1576–1590, Jun. 2018.

[85] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python Sci. Conf.*, Jul. 2015, pp. 18–24.

[86] L. Tan and J. Jiang, *Digital Signal Processing: Fundamentals and Applications*. New York, NY, USA: Academic, 2018.

[87] J. Redmon. (Oct. 15, 2016). *Darknet: Open Source Neural Networks in C*. [Online]. Available: http://pjreddie.com/darknet/

[88] AlexeyAB. *Darknet [Repository]*. Accessed: Sep. 30, 2019. [Online]. Available: https://github.com/AlexeyAB/darknet

[89] P. Goodarzi, "Comparison and optimization of CNN-based object detectors for fisheye cameras," Ph.D. dissertation, Freie Universität Berlin, Berlin, Germany, 2019.

[90] J. Redmon. (Sep. 17, 2019). *ImageNet Classification*. [Online]. Available: https://pjreddie.com/darknet/imagenet/

[91] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, Mar. 2017.

[92] R. Amami, D. Ben Ayed, and N. Ellouze, "Practical selection of SVM supervised parameters with different feature representations for vowel recognition," 2015, *arXiv:1507.06020*. [Online]. Available: http://arxiv.org/abs/1507.06020

[93] S.-C. Jou, T. Schultz, and A. Waibel, "Whispery speech recognition using adapted articulatory features," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, 2005, pp. I-1009–I-1012.

[94] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 466–481.

[95] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[96] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5561–5569.

[97] S. M. Siniscalchi, D. Yu, L. Deng, and C.-H. Lee, "Exploiting deep neural networks for detection-based speech recognition," *Neurocomputing*, vol. 106, pp. 148–157, Apr. 2013.

[98] H. Chung, J. G. Park, and H. Jung, "Rank-weighted reconstruction feature for a robust deep neural network-based acoustic model," *ETRI J.*, vol. 41, no. 2, pp. 235–241, Apr. 2019.

[99] X. Song, G. Wang, Z. Wu, Y. Huang, D. Su, D. Yu, and H. Meng, "Speech-XLNet: Unsupervised acoustic model pretraining for self-attention networks," 2019, *arXiv:1910.10387*. [Online]. Available: http://arxiv.org/abs/1910.10387

[100] T. Zoughi, M. M. Homayounpour, and M. Deypir, "Adaptive windows multiple deep residual networks for speech recognition," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112840.

MOHAMMED ALGABRI received the master's degree from King Saud University, where he is currently pursuing the Ph.D. degree with the Computer Science Department, College of Computer and Information Science. His research interests include deep learning, speech recognition, soft computing techniques, and robotics.

HASSAN MATHKOUR received the Ph.D. degree from The University of Iowa, USA. He is a currently a Professor with the Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His research interests include intelligent systems, peer-to-peer systems, modeling and analysis, database management systems, data mining, knowledge management, e-learning, and bioinformatics. He has over 100 research articles in journals and conferences. He held several administration posts including, the Dean, the Associate Dean, the Department Chair, the Director of the Research Centre, and the Head of a joint Ph.D. Program. He also serves as an IT Consultant.

MOHAMED ABDELKADER BENCHERIF received the Engineering degree in control from INELEC, Boumerdes, Algeria, in 1992, and the master's degree in signals and systems and the Ph.D. degree in the classification of remote sensing images from Saad Dahleb University, Blida, Algeria, in 2005 and 2015, respectively. He worked in diverse industrial projects, mainly on project management. He is actually working at the Centre of Smart Robotics Research, King Saud University. His research interests inlcude robotic design, speech classification, and pattern recognition.

MANSOUR ALSULAIMAN received the Ph.D. degree from Iowa State University, USA, in 1987. Since 1988, he has been with the Computer Engineering Department, King Saud University, Riyadh, Saudi Arabia, where he is currently a Professor with the Department of Computer Engineering. He is also the Director of the Centre of Smart Robotics Research, King Saud University. His research interests include automatic speech/speaker recognition, automatic voice pathology assessment systems, computer-aided pronunciation training systems, and robotics. He was the Editor-in-Chief of the *Journal of King Saud University—Computer and Information Sciences Section*.

MOHAMED AMINE MEKHTICHE was born in Medea, Algeria, in 1987. He received the B.S. degree in electronic engineering from the University of Blida, in 2010, and the M.S. degree in electronic engineering from the University of Blida, in 2012. From 2014 to 2018, he was a Researcher with the Centre of Smart Robotic Research, King Saud University, Saudi Arabia. His current research interest includes image processing stereo vision.