

창의자율과제 결과보고서

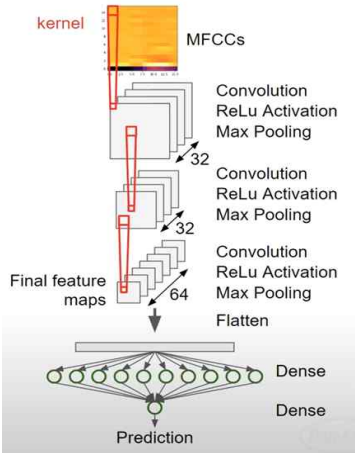
책임지도교수	성명	박정식	학부(과)	ELLT학과	직급	부교수	
공동지도교수	성명		학부(과)		직급		
참여기업/담당자							
과 제 명	자동차 음성 제어를 위한 Voice Trigger 모듈 개발						
참여학생명단 (대학원생포함)	장민지, 윤홍인, 나형주, 나희정, 임석영, 김진성, 김준하						
과제개발기간	2020년 07월 20일 ~ 2020년 11월 30일						
과제개발결과요약							
<p>- 연구목적</p> <p>차량 내 대화형 음성인식 환경에서 음성인식 구동을 제어하는 Voice Trigger 모듈이 필요하며, 본 과제를 통해 해당 모듈을 개발함.</p> <p>- 개발 결과물</p> <ul style="list-style-type: none"> • Trigger Keyword 선정 및 DB 구축: trigger keyword 모델 학습을 위하여 일반적인 데이터가 아닌 특정 단어를 정하여 음성 데이터를 직접 녹음하여 수집함. • Trigger word 모델 학습: HMM, CNN 기술을 활용하여 음향 모델을 구축함. 실험 및 성능 평가를 통하여 CNN 모델에 비하여 비교적 적은 하드웨어 용량을 필요로 하는 HMM 모델의 유효성을 검증함. • 음성 입력 및 인식 모듈 개발: 음성을 입력 받아 trigger word detection을 수행하는 모듈을 개발함. <p>- 기대효과</p> <ul style="list-style-type: none"> • 차량 내부처럼 잡음이 심하고 인식 정확도가 저하되는 환경에서, 음성인식 구동을 위한 trigger word 인식의 성능 안정화에 기여할 것으로 기대됨. • 연산량이 적은 기계학습 기법(HMM)으로도 안정된 성능을 달성함으로써 자동차 내장형 시스템에서의 직접 구동 가능성을 확인함. • 다양한 목적이 공존하는 음성인식 환경에서 업무의 도메인을 나누는 것은 태스크를 확실하면서도 간단하게 처리하기 위한 필수적인 요소이기 때문에 trigger word 인식기는 차량뿐만 아니라 다양한 환경에서 활용될 것으로 기대됨. 							
				※ 평가	A	B	C
<p style="text-align: center;">상기와 같이 창의자율과제 수행 결과를 보고합니다.</p> <p style="text-align: center;">2020년 11월 25일</p> <p style="text-align: right;">책 임 지 도 교 수 : <u>박정식</u> (인)</p> <p style="text-align: right;">인 재 양 성 부 장 : _____ (인)</p> <p>(*) 첨부:</p> <p style="text-align: center; margin-top: 20px;">자동차 기능안전 SW 연구센터장 귀하</p>							

1. 창의자율과제 내용요약 (3쪽 이상으로 작성)

가. 창의자율과제 주요내용 및 기대효과

과제 결과물 (제품, 작품, 기구 등)	<ul style="list-style-type: none"> - 기계학습을 이용한 음향 모델 구축 결과물(프로그램 소스 코드, 관련 파일) - Trigger word 인식을 위한 학습 및 테스트용 음성 데이터 수집 결과물(음성 파일) - 오프라인 인식 실험 결과(HMM 모델 정확도 97%, CNN 모델 정확도 95% 달성)
과제 (제품/재료) 내용	<ul style="list-style-type: none"> - 자동차 내부의 음성인식 구동을 위해 필요한 Trigger word 인식 모듈의 경량화에 대한 연구를 수행함. - CNN 모델보다 HMM 모델이 상대적으로 적은 연산량을 요구하며, 두 종류의 모델을 구축하고 실험을 통해 성능 결과를 비교함. - HMM 모델은 HTK를 이용하여 구현하였으며, CNN 모델은 TensorFlow를 사용하여 구현함.
사업성 및 교육효과 (고용창출연계)	<ul style="list-style-type: none"> - 본 연구를 통해, 차량 내부처럼 잡음이 심하고 인식 정확도가 저하되는 환경에서 음성인식 구동을 위한 trigger word 인식의 성능 안정화를 확인함. - 연산량이 적은 기계학습 기법(HMM)으로도 안정된 성능을 달성함으로써 자동차 내장형 시스템에서의 직접 구동 가능성을 확인함. - 이 같은 연구 결과를 통해 본 연구는 음성 기반 차량용 지능형 인터페이스 연구의 기초 연구 자료로 활용될 수 있고, 나아가 지능형 자동차 기술 사업화에 기여할 수 있을 것으로 예상됨.

나. 창의자율과제 설계 과정 설명 (사진을 최대한 활용하여 작성)

<p>[과정1]</p> <pre>#5. 학습을 위한 폴더 생성 os.system("mkdir hmm0") #6. hmmdefs, macros 파일 만들기 os.chdir(file_path) print(os.getcwd()) os.system("HCompV -C configtr -f 0.01 -m -S train.scp -M hmm0 proto") #7. hmm1-5까지 파일 생성 후 hmmdefs, macros 만들기 #8. words.mlf, wlist 확인 후 실행 os.chdir(file_path) print(os.getcwd()) os.system("mkdir hmm1") os.system("HERest -C configtr -l words.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 wlist") os.chdir(file_path) print(os.getcwd()) os.system("mkdir hmm2") os.system("HERest -C configtr -l words.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm1/macros -H hmm1/hmmdefs -M hmm2 wlist") os.system("mkdir hmm3") os.system("HERest -C configtr -l words.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm2/macros -H hmm2/hmmdefs -M hmm3 wlist") os.system("mkdir hmm4") os.system("HERest -C configtr -l words.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm3/macros -H hmm3/hmmdefs -M hmm4 wlist") os.system("mkdir hmm5") os.system("HERest -C configtr -l words.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm4/macros -H hmm4/hmmdefs -M hmm5 wlist") #9. test.scp, dict 확인하기 os.chdir(file_path) print(os.getcwd()) os.system("hVite -C configtr -H hmm5/macros -H hmm5/hmmdefs -S test.scp -i recout.mlf -w wdict -n 5 5 dict wlist") #10. 결과 파일과 비교해서 정확도 비교하기 os.chdir(file_path) os.system("HResults -l words.mlf wlist recout.mlf > result_keyword.txt")</pre>	<p>[과정설명]</p> <p>HMM 모델을 구현하는 코드. hmmdefs과 macros 파일로 학습 모델을 만들어 트레이닝을 거듭하여 가장 좋은 결과 값을 찾는 코드. HTK의 HERest 명령어를 사용하여 학습이 끝나면 test파일을 input값으로 하여 실험한 후 결과를 확인함.</p>
<p>[과정2]</p>  <pre>#Build model #Based on Keras model = models.Sequential() model.add(layers.Conv2D(32, (2,2), activation = 'relu', input_shape = sample_shape)) model.add(layers.MaxPool2D(pool_size=(2,2))) model.add(layers.Conv2D(32,(2,2),activation = 'relu')) model.add(layers.MaxPool2D(pool_size=(2,2))) model.add(layers.Conv2D(64,(2,2),activation = 'relu')) model.add(layers.MaxPool2D(pool_size=(2,2))) #Classifier model.add(layers.Flatten()) model.add(layers.Dense(64,activation='relu')) model.add(layers.Dropout(0.5)) model.add(layers.Dense(1,activation='sigmoid'))</pre>	<p>[과정설명]</p> <p>CNN 구현 코드 및 그림. 모델을 만드는 layer 부분과 분류하는 classifier 부분으로 나뉨. layer 단계에서는 mfcc에서 특징을 추출하고 특징벡터를 학습함. 이 학습 결과를 토대로 classifier에서는 분류 작업을 함. input이 들어오면 그게 모델과 같은 음성인지 아닌지를 확인하여 결과 값을 내는 단계.</p>

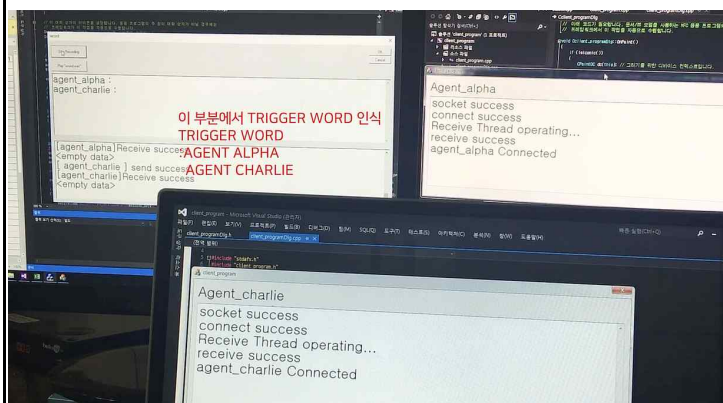
※창의자율과제 설계 과정 실물사진 3매 이상 첨부

다. 최종결과물(제품, 작품, 기구 등) 사진

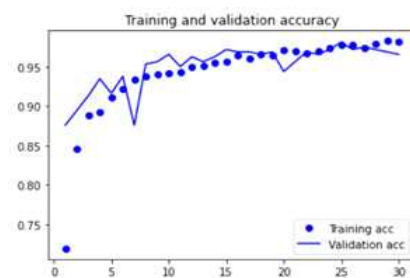
		실험 1. HTK		실험 2. HTK		실험 3. CNN	
TRIGGER WORD		'Hey Lumos'		'Agent Alpha', 'Agent Charlie'		'Backward'	
GARBAGE		'PBWdata1'		'PBWdata1'		'cat','dog','eight','five'	
전체 데이터		400		1000		3678	
TRIGGER WORD	GARBAGE	200	200	500	500	1664	1664
Train	Test or Validation	360	40	800	200	2996	332
인식결과		100%		97%		95%	
False Acceptance Rate	False Rejection Rate	0%	0%	2.1%	0.9%	0.02%	0.01%

<실험 1,2,3의 데이터 수와 인식 결과>

- 실험1과 실험2는 실제 음성 데이터를 녹음 후 수집한 데이터를 활용하여 HMM 모델을 구축하고 이를 평가한 실험 결과이며, 실험3은 google command dataset을 이용하여 CNN 모델을 구축하고 이를 평가한 결과임.
- 전체 데이터의 80%를 학습에 사용하였고 나머지 20%를 htk에서는 test set으로, CNN에서는 10%를 test, 10%를 validation set으로 사용함.
- 실험 결과, 비교적 적은 연산량을 요구하는 HMM의 경우 실험1에서 인식을 100%를 달성하였고, CNN의 경우 95%의 정확도를 나타냄.
- 실험2의 경우, 실험1과 달리 두 개의 trigger word를 사용하여 실험1보다 다소 낮은 성능을 보였으나, CNN 모델과 비교되는 수준의 성능(97%)으로 유의미한 결과를 보임.



<실험 2에서 만든 모델로 실제 성능 실험>



<실험 3에서의 Accuracy 결과>