Abstract

이 논문에서 풀려고 한 문제
이미지 기반 시퀀스 인식에서 가장 중요한 scene text recognition

이전 아키텍처와 차이점
1. End to end
2. Character segmentation, horizontal scale normalization 사용 X, 임의의 길이로 시퀀스를 자연스럽게 처리
3. vocab에만 국한되지 않음
4. 효과적이면서 작은 모델

악보 인식

Introduction

배경
DCNN 이후 논문, 신경망의 부활 목격중
그러나 detection과 classification에만 쓰이고 있다.

당시 주요 논문:
[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 1, 3 (RCNN)
[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012. 1, 3 (AlexNet)

우리는 이미지 기반 시퀀스 인식(image based sequence recognition)에 초점을 맞춘다.(실제 세계와 유사)

단일 레이블이 아니라 시퀀스 레이블을 예측해야하는 경우 종종 발생

시퀀스 레이블 특징, DCNN 한계
길이가 엄청 다양할 수 있다.
DCNN은 고정길이를 가지기 때문에 가변적인 시퀀스를 다룰 수 없다.

DCNN:
[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012. 1, 3
[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient- based learning applied to document recognition. *Proceed- ings of the IEEE*, 86(11):2278–2324, 1998. 1

우리와 같은 시도가 있긴 했다.
- character를 하나하나 detect하고 DCNN모델에 넣는 방식. (label이 지정된 문자 이미지를 사용하여 훈련됨)
문제점 : 각 문자를 정확하게 detect하고 잘라내기 위해 강력한 character detector가 필요
- 이미지 분류 문제로 접근해 word별로 label할당 (총 90K개 단어)
문제점 : 중국어 텍스트, 악보같은 다른 유형의 시퀀스 유사 개체로 일반화되기 어려움
=> 정리 : DCNN으로는 해결할 수 없다.

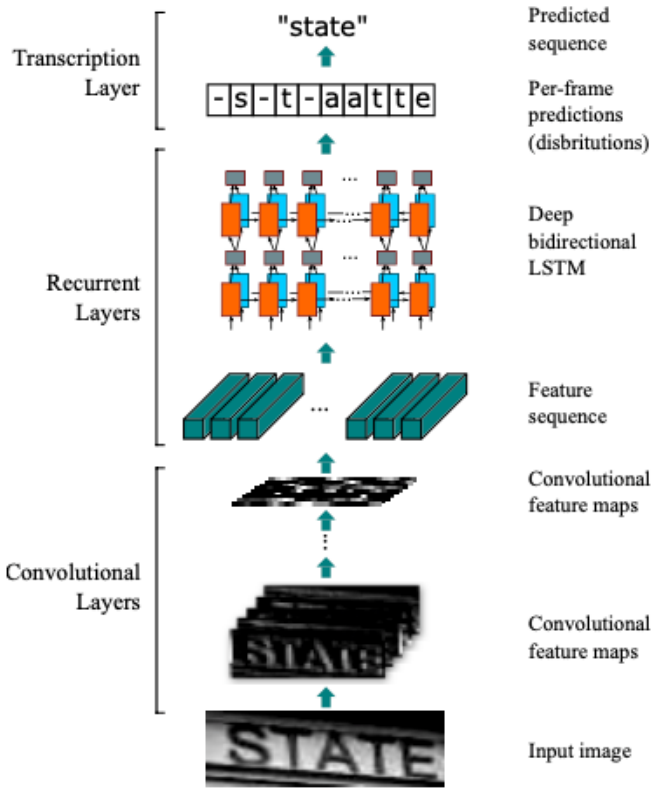
시도:
- 8,35
- 22

RNN
장점 : 훈련이나 테스트나 이미지의 각 요소의 위치가 필요하지 않다.
문제점 : 그러나 전처리단계에서 이미지를 이미지 피쳐의 시퀀스로 변환해야함
=> 이건 end-to-end가 될 수 없다.

신경망이 아닌 전통적 scene text recognition
Mid-level feature 사용 -> 일반적으로 성능 우수

우리 논문
새로운 아키텍처, CRNN, DCNN과 RNN의 결합, 특색있는 장점을 가진다.
1. 시퀀스 레이블에서 직접 학습이 가능하고 자세한 annotation이 필요없다.
2. Handcraft features나 binarization, segmentation, component localization같은 전처리가 필요없다.
3. RNN처럼 라벨의 시퀀스를 생성할 수 있다.
4. 길이 제한이 없고 높이 정규화만 하면 된다.
5. 성능 좋다.
6. 모델 크기도 DCNN보다 작다.

아키텍처



3개의 컴포넌트
- 아래부터 convolutional layer, recurrent layer, transcription layer

Convolutional layer
- 자동으로 feature sequence를 추출

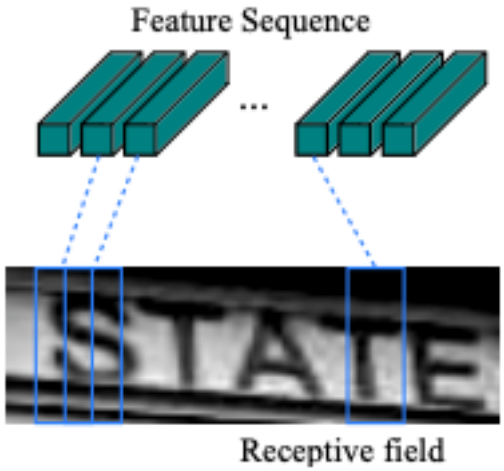
Recurrent layer
- 각 프레임 예측

Transcription layer
- 프레임 당 예측을 시퀀스로 변환

CRNN은 CNN과 RNN처럼 다른 종류의 아키텍처로 구성되지만 하나의 loss function으로 훈련될 수 있다.

2.1 Feature Sequence 추출

CNN에서 FC layer를 제거하고 convolutional, max pooling layer를 가져옴
이미지 높이는 동일한 높이로 맞춰줘야한다.
Feature map을 output으로 뽑아 Recurrent layer의 input으로 사용
Feature sequence의 각 feature vector는 feature map상의 좌->우로 생성됨
i번째 feature vector는 모든 map의 i번째 컬럼과 concat된다는 뜻
우리 세팅에서 각 컬럼의 width는 한 픽셀(single pixel)로 정해진다.



2.2 Sequence 라벨링

RNN의 장점 3가지

1. 시퀀스 내에서 contextual(상황에 맞는) 정보를 캡쳐하는 강력한 기능을 가진다.
   시퀀스로 다루는게 더 stable하고 도움이된다.
   -> ex) i와 l을 각각 구분하는 것보다 문맥 상에서 높이를 구분하며 인식하는 것이 더 구분하기 쉽다.
2. RNN은 오류차이를 convolution layer로 역전파할 수 있기 때문에 recurrent layer와 convolution layer를 동시에 훈련시킬 수 있다.
3. RNN은 임의의 길이의 시퀀스에서 작동하여 처음부터 끝까지 이동할 수 있다.
   RNN 문제 : vanishing gradient problem으로 저장할 수 있는 context 범위 제한, 훈련 프로세스 부담 증가
   LSTM 장점 : 과거 context 저장하고 forget또한 가능, longterm dependency 해결
   LSTM 문제 : 방향성이 있으며 과거 context만 사용함. 이미지에서는 양방향이 필요
   그래서 biLSTM 사용

=> convolution layer와 recurrent layer 사이 다리역할을 하는 map to sequence라는 사용자지정 네트워크 layer 만듬

2.3 Transcription

예측을 시퀀스로 변환하는 과정

Lexicon-free/ lexicon-based 방식이 존재
lexicon-free에서는 lexicon 없이 예측
lexicon-based는 확률이 가장 높은 시퀀스를 선택 예측

2.3.1 probability of label sequence
CTC를 채택함

2.3.2 Lexicon-free transcription
각 타임스탬프에서 확률이 가장 큰 레이블이 사용되고 근사치로 결과 시퀀스가 매핑됨

2.3.3 Lexicon-based transcription
가장 확률이 높은 라벨 시퀀스를 선택
모든 시퀀스에 대해 1번 식을 계산하고 확률이 높은 결과를 선택하는데에는 시간이 많이 걸림

$$p(\mathbf{l}|\mathbf{y}) = \sum_{\boldsymbol{\pi}:\mathcal{B}(\boldsymbol{\pi})=\mathbf{l}} p(\boldsymbol{\pi}|\mathbf{y}), \tag{1}$$

$$\mathbf{l}^* = \arg\max_{\mathbf{l}\in\mathcal{N}_\delta(\mathbf{l}')} p(\mathbf{l}|\mathbf{y}). \tag{2}$$

Lexicon-free 에서 nearest neighbor candidates로 검색을 제한, BK tree를 활용

2.4 Network Training

$$\mathcal{O} = -\sum_{I_i,\mathbf{l}_i\in\mathcal{X}} \log p(\mathbf{l}_i|\mathbf{y}_i), \tag{3}$$

(Cross entropy 같은데?)
End to end가 가능하다
네트워크는 SGD로 훈련되고 양방향 역전파, BPTT를 적용함
Optimization은 ADADELTA를 사용함. 학습 속도를 수동으로 설정할 필요가 없다.
=> ADADELTA가 모멘텀 방법보다 더 빨리 수렴한다!

Experiments

평가하기 위해 standard benchmark랑 악보인식을 진행함

3.1 training, testing을 위한 dataset 설정

3.2 scene text image에 대한 CRNN의 자세한 설정
- VGG-VeryDeep 3,4번째 max pooling에서는 1x2사이즈 pooling window 사용

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512, k:2 × 2, s:1, p:0
MaxPooling	Window:1 × 2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
MaxPooling	Window:1 × 2, s:2
Convolution	#maps:256, k:3 × 3, s:1, p:1
Convolution	#maps:256, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:128, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:64, k:3 × 3, s:1, p:1
Input	$W \times 32$ gray-scale image

we adopt 1 × 2 sized rectangular pooling windows instead of the conventional squared ones. This tweak yields feature maps with larger width, hence longer feature sequence. For example, an image containing 10 characters is typically of size 100 × 32, from which a feature sequence 25 frames can be generated.

- 5,6 번째 convolution layer 뒤에 BN삽입 -> 훈련 프로세스 가속화
- LSTM (torch)
- transcription (C++)
- BK tree (C++)
- 네트워크 ADADELTA로 훈련, 파라미터 ρ = 0.9
- 훈련 중 모든 이미지는 100x32
- 테스트 영상은 높이 32, 너비는 높이에 비례하여 조정되지만 최소 100 이상
- 인텔 Xeon E5-2609 CPU, 64GB RAM, Tesla K40 GPU
- Lexicon-free, IC03 기준 테스트 시간 : 샘플 당 평균 0.16초
- Lexicon-based, IC03 기준 테스트 시간 : 평균 0.53초

3.3 종합적인 비교 결과
- lexicon-free도 가능하기때문에 전화번호같은 임의의 문자열에도 강건하고 문장, 중국어 단어 같은 다른 스크립트도 처리할 수 있다.
- word-level을 기준으로 훈련, PhotoOCR은 790만 개의 character-level을 기준으로 훈련함
- 우리모델은 다른 것과 다르게 lexicon에 의존하지 않는다.

	E2E Train	Conv Ftrs	CharGT-Free	Unconstrained	Model Size
Wang *et al.* [34]	✗	✗	✗	✔	-
Mishra *et al.* [28]	✗	✗	✗	✗	-
Wang *et al.* [35]	✗	✔	✗	✔	-
Goel *et al.* [13]	✗	✗	✔	✗	-
Bissacco *et al.* [8]	✗	✗	✗	✔	-
Alsharif and Pineau [6]	✗	✔	✗	✔	-
Almazán *et al.* [5]	✗	✗	✔	✗	-
Yao *et al.* [36]	✗	✗	✗	✔	-
Rodrguez-Serrano *et al.* [30]	✗	✗	✔	✗	-
Jaderberg *et al.* [23]	✗	✔	✗	✔	-
Su and Lu [33]	✗	✗	✔	✔	-
Gordo [14]	✗	✗	✗	✗	-
Jaderberg *et al.* [22]	✔	✔	✔	✗	490M
Jaderberg *et al.* [21]	✔	✔	✔	✔	304M
CRNN	✔	✔	✔	✔	**8.3M**

E2E Train
- end to end 훈련이 되는가?

Conv Ftrs
- Convolution 기능을 직접 사용하는가? (x면 handcraft)

CharGT-Free
- Character-level annotation으로부터 자유로운가?

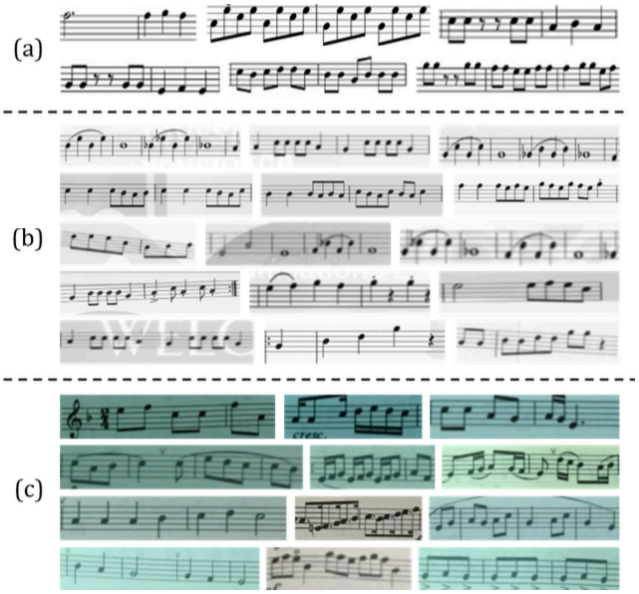
Unconstrained
- lexicon 외의 단어도 예측이 가능한가?

Model Size
- 모델의 사이즈

	IIIT5k			SVT		IC03				IC13
	50	**1k**	**None**	**50**	**None**	**50**	**Full**	**50k**	**None**	**None**
ABBYY [34]	24.3	-	-	35.0	-	56.0	55.0	-	-	-
Wang *et al.* [34]	-	-	-	57.0	-	76.0	62.0	-	-	-
Mishra *et al.* [28]	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-
Wang *et al.* [35]	-	-	-	70.0	-	90.0	84.0	-	-	-
Goel *et al.* [13]	-	-	-	77.3	-	89.7	-	-	-	-
Bissacco *et al.* [8]	-	-	-	90.4	78.0	-	-	-	-	87.6
Alsharif and Pineau [6]	-	-	-	74.3	-	93.1	88.6	85.1	-	-
Almazán *et al.* [5]	91.2	82.1	-	89.2	-	-	-	-	-	-
Yao *et al.* [36]	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-
Rodrguez-Serrano *et al.* [30]	76.1	57.4	-	70.0	-	-	-	-	-	-
Jaderberg *et al.* [23]	-	-	-	86.1	-	96.2	91.5	-	-	-
Su and Lu [33]	-	-	-	83.0	-	92.0	82.0	-	-	-
Gordo [14]	93.3	86.6	-	91.8	-	-	-	-	-	-
Jaderberg *et al.* [22]	97.1	92.7	-	95.4	80.7*	**98.7**	**98.6**	93.3	**93.1***	**90.8***
Jaderberg *et al.* [21]	95.5	89.6	-	93.2	71.7	97.8	97.0	93.4	89.6	81.8
CRNN	**97.6**	**94.4**	**78.2**	**96.4**	**80.8**	**98.7**	97.6	**95.5**	89.4	86.7

δ(델타) 의 역할
- 인식 정확도 results가 클수록 후보가 많이 생김, 정확도 높아짐, BK트리 검색시간 길어짐, 계산비용 커짐
-> δ = 3으로 사용

3.4 일반성 입증을 위한 악보인식에의 적용



- Pitches(음 높이)만 인식, 코드 무시, 모두 동일한 C장조를 가정함
- 데이터셋 이미지 2650개 수집
- 이미지(fragment) 별 3~20개의 음표 존재
- 일부러 augmentation 줘서 손상주고 265k개의 train data 만듬
- a = clean, b = 합성된, c = 실제 세계

성능 평가
1. 올바르게 인식한 비율
2. Ground truth와 predicted pitch sequence 사이의 average edit distance

=> 상용 엔진인 Photoscore, capella scan보다 성능 월등히 좋다.

주된 이유 :
오선지를 감지하기 위한 강력한 binarization과정에 의존하지만 조명, 손상, 배경 등의 문제로 binarization에 실패하는 경우가 많다.
그러나 CRNN은 소음과 왜곡에있어 매우 강건하다.
심지어 contextual information을 활용할 수 있어 주변 음과 비교하여 인식할 수 있다.

Conclusion

CRNN 정리

- 다양한 차원의 입력이미지 받을 수 있다.
- 다른 길이의 예측 생성할 수 있다.
- word-level에서 실행된다.
- 자세한 annotation이 필요없다.
- FC layer를 버려서 더 작고 효율적인 모델이다.

=> 이 속성들로 인해 CRNN은 image-based sequence recognition에 있어 excellent approach다.

- 성능이 좋다.
- 다른 도메인에서도 사용 가능하다.