Fully Convolutional Networks for Handwriting Recognition

Felipe Petroski Such*, Dheeraj Peri*,
Frank Brockler†, Paul Hutkowski†,
Raymond Ptucha*
*Rochester Institute of Technology,
†Kodak Alaris,

R·I·T

Kodak alaris

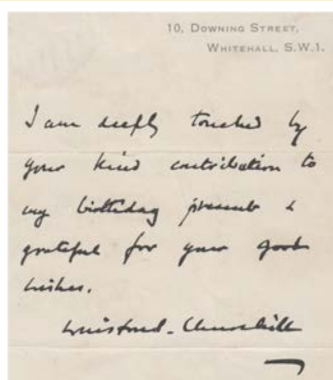
ICFHR 2018

The 16th International Conference on Frontiers in Handwriting Recognition

August 5 - 8, 2018 • Niagara Falls, USA

Background

- Offline handwriting recognition continues to be a difficult process due to the virtually infinite ways the same information can be written.
- Convolutional Neural Networks (CNNs) and have been applied to handwriting recognition with good success.
- Recurrent Neural Networks (RNNs) are useful for arbitrary length sequences and Connectionist Temporal Classification (CTC) are good as a post correction step.


10, DOWNING STREET,
WHITEHALL, S.W.1.

```
I am truly touched by
your kind contribution to
my birthday presents &
grateful for your good
wishes.
Winston Churchill
```

Note: Some believe the above letter is a forgery.

Workflow- Word Extraction

Document Segmentation | Block Segmentation

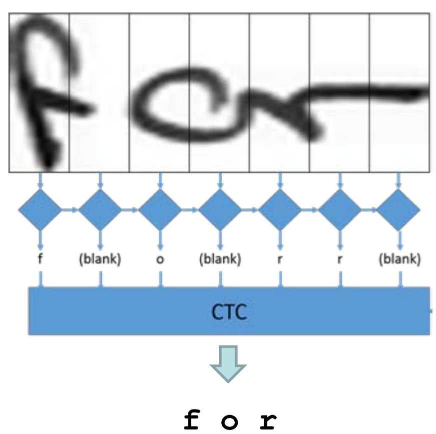
SegNet or similar labels each pixel by type- can grow to orthogonal boundaries.

Modified XY Tree or similar suggests rectilinear splits.

Use both to define paragraphs, sentences and word blocks.

Workflow- Word Recognition

- Preprocessing
  - Fix skewing, rotation, contrast
- Prediction
  - CNNs, HMM, LSTMs used together
- Post-processing
  - Train & Test: CTC
  - Test: Language Model



Proposed Method

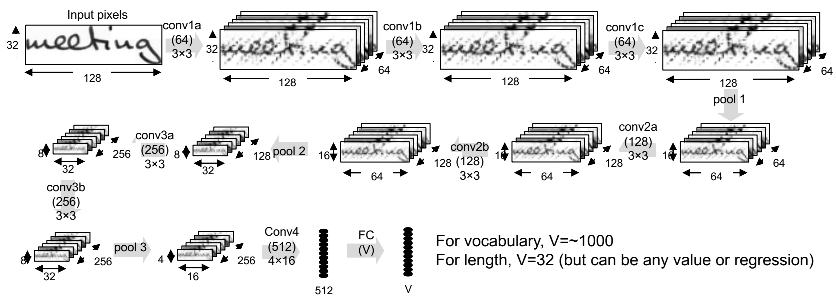
- Character classification without the need for:
  - Preprocessing- no deskewing
  - Predefined lexicon of words- can work on surnames, phone numbers, and street addresses
  - Post processing- No RNN or CTC needed
- Utilizes Fully Convolutional Networks (FCNs) to translate arbitrary sequence length.
  - FCNs are faster to train than RNNs and more robust
  - CTC can still be used, but we found them hard to converge
- Single architecture works on arbitrary words as well as words from a lexicon

High Level

Vocabulary CNN → Length CNN → Symbol CNN → Language Model

Vocabulary CNN	Length CNN	Symbol CNN	Language Model
CNN Predicts word label for common words such as '**`his`**', '**`her`**', '**`the`**'. If confidence > γ, then done!	CNN Predicts the number of symbols, then resample block to 32 X 16*N*, where *N* is the number of symbols.	FCN Predicts 2*N*+1 symbols, where each symbol is separated by a blank space.	(optional step) When block is known to come from a lexicon of words, use vocabulary matching by minimizing character error rate.

Vocabulary and Length CNNs



C(64,3,3)-C(64,3,3)-C(64,3,3)-P(2)-C(128,3,3)-C(128,3,3)-C(256,3,3)-P(2)-C(256,3,3)-C(512,3,3)-C(512,3,3)-P(2)-C(256,4,16)-FC(V)-SoftMax where C(D,H,W) stands for convolution with the dimensions of the filter as H╳W and the depth D. Each convolutional layer is followed by a batch norm and ReLU layer. P(2) represents a 2 ╳ 2 pooling layer with stride 2.

High Level

Vocabulary CNN ⇨ Length CNN ⇨ Symbol CNN ⇨ Language Model

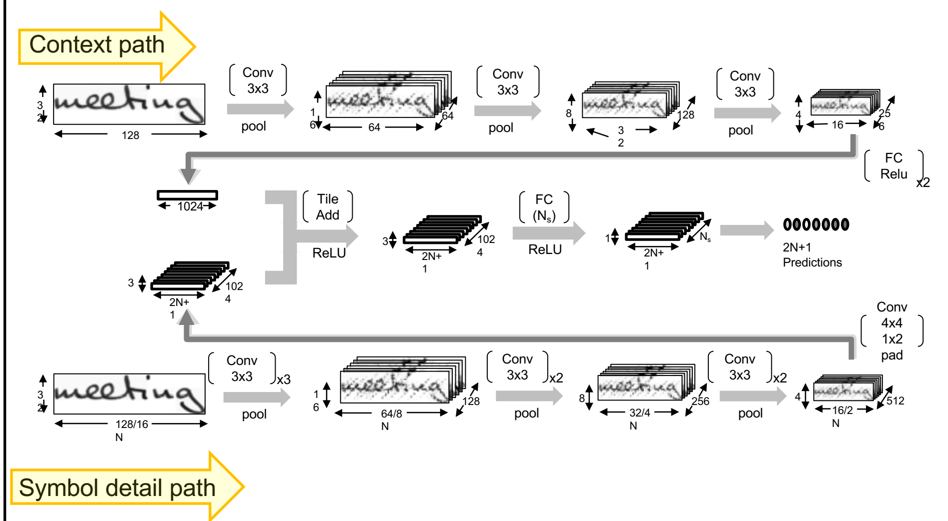
Vocabulary CNN: CNN Predicts word label for common words such as '`his`', '`her`', '`the`'. If confidence > γ, then done!

Length CNN: CNN Predicts the number of symbols, then resample block to 32 ╳ 16*N*, where *N* is the number of symbols.

Symbol CNN: FCN Predicts 2*N*+1 symbols, where each symbol is separated by a blank space.

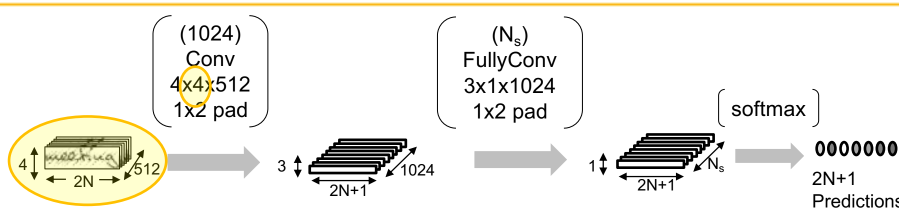
Language Model: (optional step) When block is known to come from a lexicon of words, use vocabulary matching by minimizing character error rate.

Symbol FCN

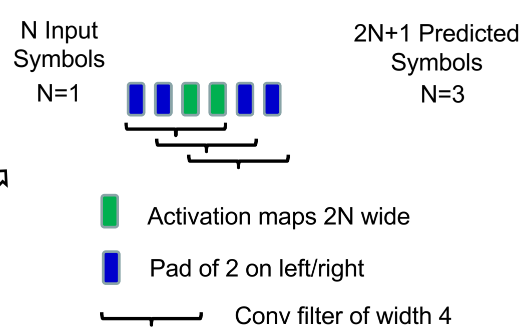


Such et al. ICFHR'18 11

Symbol FCN

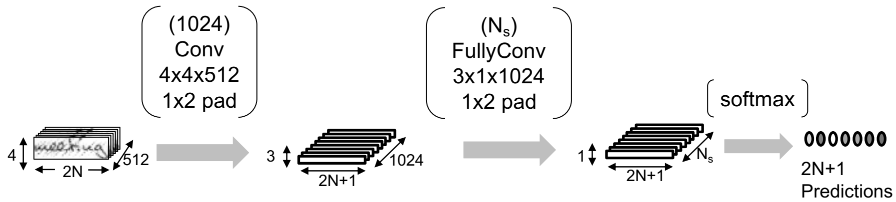


- Vertical pad gives forgiveness for up/down- can think as three estimates for each prediction.
- Horizontal pad gives 2N+1 outputs.

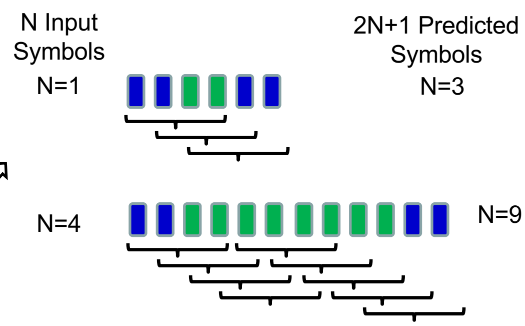


Such et al. ICFHR'18 12

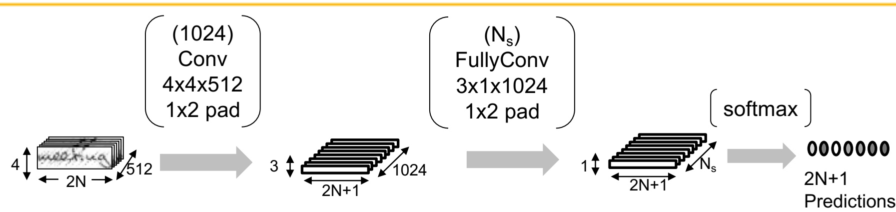
Symbol FCN



- Vertical pad gives forgiveness for up/down- can think as three estimates for each prediction.
- Horizontal pad gives 2N+1 outputs.



Symbol FCN



- Vertical pad gives forgiveness for up/down- can think as three estimates for each prediction.
- Horizontal pad gives 2N+1 outputs.

- Each of 2N+1 predictions are a linear combination of 3x1024 activation map.

- Softmax over N_s symbols.

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	-	-	-	-	-
i	2	-	-	-	-	-
m	3	-	-	-	-	-
e	4	-	-	-	-	-

Such et al. ICFHR'18

Predicted Word

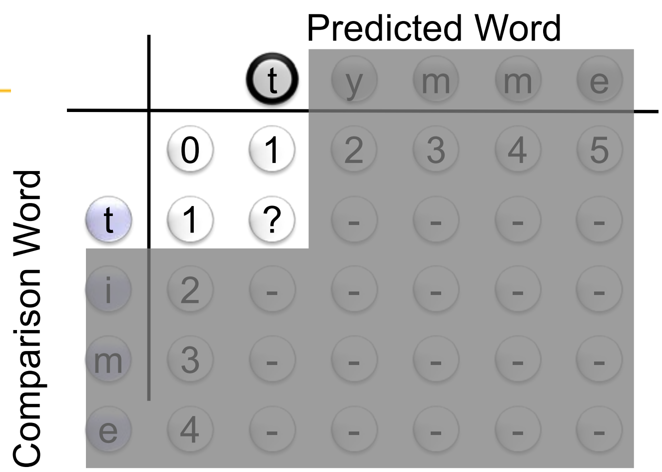
Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	?	-	-	-	-
i	2	-	-	-	-	-
m	3	-	-	-	-	-
e	4	-	-	-	-	-

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

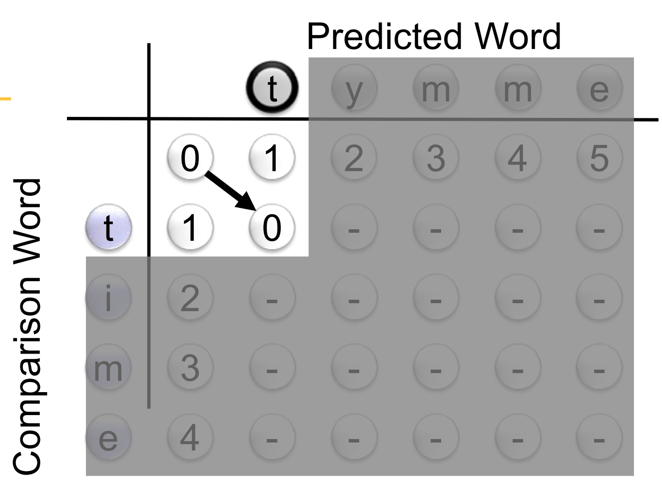


$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

17



Match! Pass along previous error

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

18

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	-	-	-	-
i	2	1	-	-	-	-
m	3	-	-	-	-	-
e	4	-	-	-	-	-

Miss!
+1 To insert i

$$C_{i,j} = min\left(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag\right)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

19

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	-	-	-	-
i	2	1	-	-	-	-
m	3	2	-	-	-	-
e	4	3	-	-	-	-

Miss!
+1 To insert m, then e

$$C_{i,j} = min\left(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag\right)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

20

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	-	-	-
i	2	1	-	-	-	-
m	3	2	-	-	-	-
e	4	3	-	-	-	-

Miss, +1 to delete y

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

21

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	-	-	-
i	2	1	1	-	-	-
m	3	2	-	-	-	-
e	4	3	-	-	-	-

Miss, +1 to replace y with i

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

22

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	-	-	-
i	2	1	1	-	-	-
m	3	2	2	-	-	-
e	4	3	-	-	-	-

Miss, +1 to replace y with m or +1 to insert m

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

23

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	-	-	-
i	2	1	1	-	-	-
m	3	2	2	-	-	-
e	4	3	3	-	-	-

Miss, +1 to replace y with e or +1 to insert e

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

24

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	-	-
i	2	1	1	-	-	-
m	3	2	2	-	-	-
e	4	3	3	-	-	-

Miss, +1 to delete m

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

25

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	-	-
i	2	1	1	2	-	-
m	3	2	2	-	-	-
e	4	3	3	-	-	-

Miss, +1 to replace m with i or +1 to delete y

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

26

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	-	-
i	2	1	1	2	-	-
m	3	2	2	1	-	-
e	4	3	3	-	-	-

Match

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

27

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	-	-
i	2	1	1	2	-	-
m	3	2	2	1	-	-
e	4	3	3	2	-	-

Miss,
Cost +1
to insert
e

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

28

Predicted Word

Comparison Word		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	3	-
i	2	1	1	2	3	-
m	3	2	2	1	-	-
e	4	3	3	2	-	-

Miss,
Cost +1
to delete
m

$$C_{i,j} = min\left(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag\right)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR‘18

29

Predicted Word

Comparison Word		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	3	-
i	2	1	1	2	3	-
m	3	2	2	1	2	-
e	4	3	3	2	-	-

Match

$$C_{i,j} = min\left(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag\right)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR‘18

30

Predicted Word

Comparison Word		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	3	-
i	2	1	1	2	3	-
m	3	2	2	1	2	-
e	4	3	3	2	2	-

Miss,
Cost +1 to replace m with e

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

31

Predicted Word

Comparison Word		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	3	4
i	2	1	1	2	3	4
m	3	2	2	1	2	3
e	4	3	3	2	2	-

Miss,
Cost +1 to delete e

$$C_{i,j} = min(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Such et al. ICFHR'18

32

Predicted Word

Comparison Word

		t	y	m	m	e
	0	1	2	3	4	5
t	1	0	1	2	3	4
i	2	1	1	2	3	4
m	3	2	2	1	2	3
e	4	3	3	2	2	2

Match

Final Cost

$$C_{i,j} = min\left(C_{i-1,j} + 1, C_{i,j-1} + 1, Diag\right)$$

$$Diag = \begin{cases} C_{i-1,j-1} & if\ pred\ char = compare\ char \\ C_{i-1,j-1} + 1 & o/w \end{cases}$$

Datasets

- IAM English handwritten dataset
  - 115,320 English words, mostly cursive, by 500 authors.
  - Comes with train, validation, test splits.
- RIMES French handwritten dataset
  - 60,000 French words by over 1,000 authors.
  - Use ICDAR2011 release and splits
- NIST Handprinted and Forms database
  - 810,000 characters by 3,600 authors

IAM Results

Model	WER	CER
Dreuw et al. [10]	18.8	10.1
Boquera et al. [11]	15.5	6.90
Kozielski et al. [18]	13.30	5.10
Bluche et al. [5]	11.90	4.90
Doetsch et al. [9]	12.20	4.70
Our work	8.71	4.43
Voigtlaender et al. [30]	9.3	3.5
Poznanski and Wolf [23]	**6.45**	**3.44**

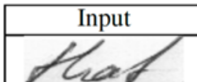
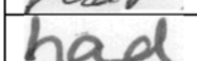
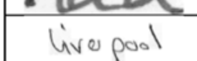
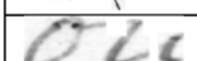
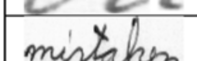
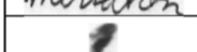
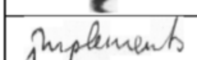
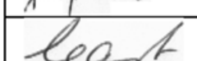
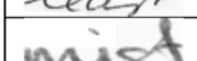
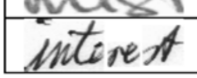
HMMS with MLP
HMMS with MLP
HMM
CNN with RNN
LSTM w/ CTC
☆
CNN w/ RNN
CNN with pre and post processing, fixed symbol lexicon of only upper and lower case Latin alphabet

☆ (our work):
Vocabulary CNN of 1100 words
Symbol CNN uses N_s=123 symbols

Such et al. ICFHR'18

35

IAM Results

Input	Label	Prediction
	that	that
	had	had
	Liverpool	livepool
	on	oui
	mistaken	mistahon
	’	,
	implements	implement
	least	least
	mist	mist
	interest	interest

36

RIMES Results

Model	WER	CER
Kozielski et al. [18]	13.70	4.60
Doetsch et al. [9]	12.90	4.30
Bluche et al. [5]	11.80	3.70
Our work	5.68	2.22
Poznanski and Wolf [23]	**3.90**	**1.90**

HMM

LSTM w/ CTC

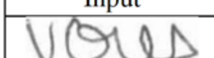
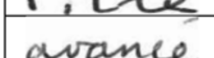
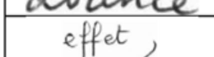
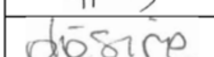
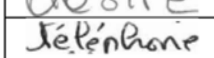
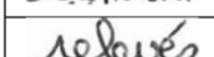
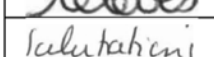
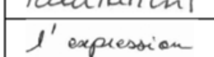
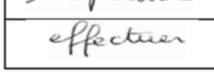
CNN with RNN

☆

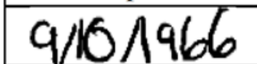
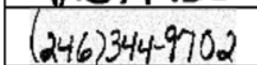
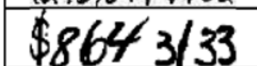
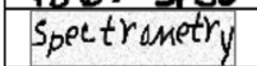
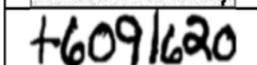
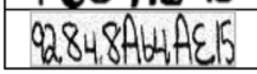
CNN with pre and post processing, fixed symbol lexicon of only upper and lower case Latin alphabet

☆ (our work):
Vocabulary CNN of 800 words
Symbol CNN uses N_s=123 symbols

RIMES Results

Input	Label	Prediction
	vous	vous
	titre	titre
	avancé	avance
	effet ,	effett
	désire	diésiire
	téléphone	télénhone
	relevés	relves
	salutations	salutations
	l'expression	l'expression
	effectuer	effectuer

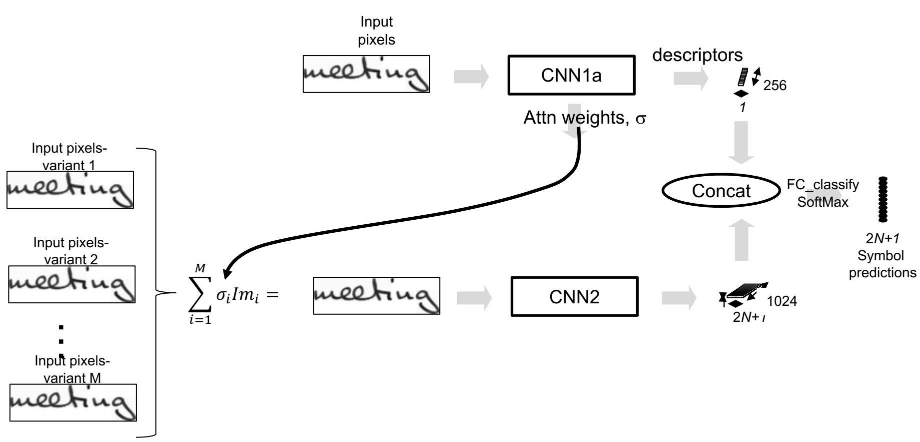
NIST Results

Input	Label	Prediction
	9/10/1966	9/10/1966
	(246)344-9702	(246)344-9702
	$8643133	$8643133
	Spectrometry	Spectrometry
	+6091620	+6091620
	92.84.8A.b4.AE.15	92.84.8A.b4AE.15

92.4% accuracy on a subset of 12,000 word blocks (English, French, and special characters) generated from NIST dataset

Such et al. ICFHR'18

Attention Modeling



Such et al. ICFHR'18

Conclusions

- Introduction of offline handwritten recognition architecture which works with either arbitrary characters or fixed lexicon.
- Vocabulary CNN quickly solves simple words.
- Length CNN forms canonical word suitable for input into Symbol CNN.
- Symbol CNN is a FCN which is indifferent to canonical word length.
- Despite using large character lexicon (123 symbols) and being able to predict arbitrary words such as surnames and phone numbers, generates competitive CER and WER.

Thank you!!

Ray Ptucha
rwpeec@rit.edu

https://www.rit.edu/mil