

CAP 6412 Advanced Computer Vision

<http://www.cs.ucf.edu/~bgong/CAP6412.html>

Boqing Gong

March 03, 2016

Next week: Spring break

The week after next week: Vision and language

<p>Tuesday (03/15)</p> <p>Fareeha Irfan</p>	<p>[Book2Movie] Tapaswi, Makarand, Martin Bauml, and Rainer Stiefelhagen. "Book2movie: Aligning video scenes with book chapters." In <i>Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition</i>, pp. 1827-1835. 2015.</p> <p>& Secondary papers</p>
<p>Thursday (03/17)</p> <p>Shreyas Somashekar</p>	<p>[Visual Genome] Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen et al. "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations." <i>arXiv preprint arXiv:1602.07332</i> (2016).</p> <p>& Secondary papers</p>

Assignment 8: Due on 03/15, 12pm

- 1. Read the following paper.
- 2. Write down the training & test algorithms described in the paper, using the template on Pages 7 & 8 of <http://www.cs.ucf.edu/~bgong/CAP6412/lec10.pdf>.

[**Book2Movie**] Tapaswi, Makarand, Martin Bauml, and Rainer Stiefelhagen. "Book2movie: Aligning video scenes with book chapters." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1827-1835. 2015.

Assignment 9: Due on 03/17, 12pm

- Review the following paper using the Paper Review Template at <http://www.cs.ucf.edu/~bgong/CAP6412/Review.docx>.

[Visual Genome] Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen et al. "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations." *arXiv preprint arXiv:1602.07332* (2016).

Today

- Administrivia
- Recurrent Neural Networks (RNNs) (II)
- OCR in the wild, by Aisha

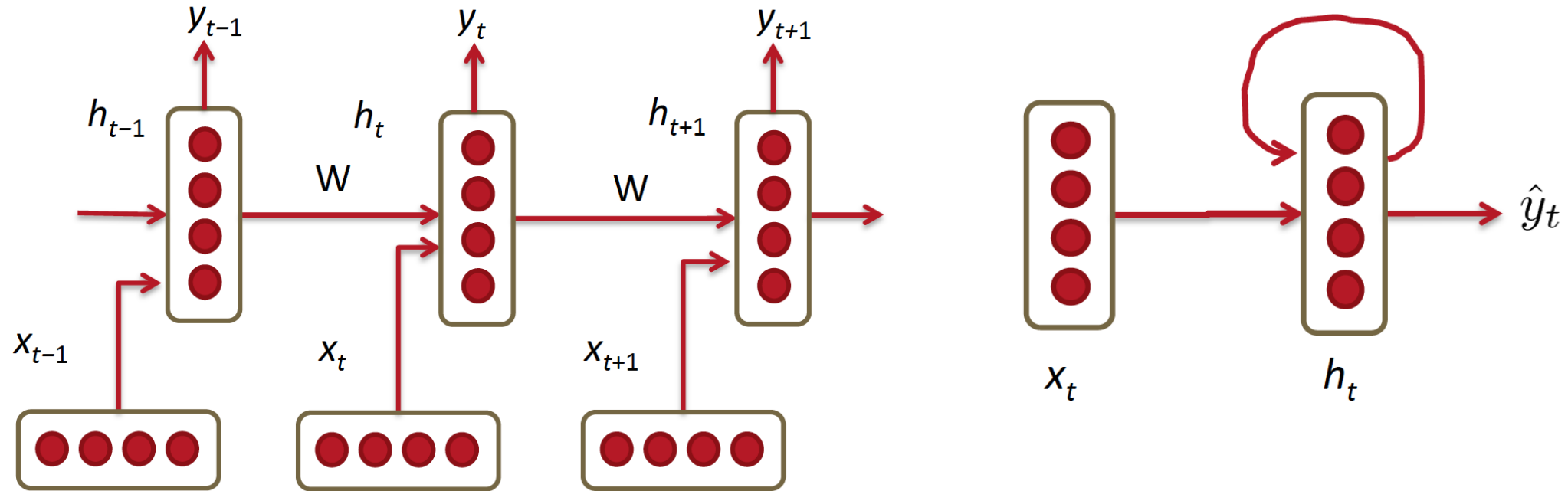
(Discrete-time) RNN

- Three time steps and beyond

$$x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots, x_T$$

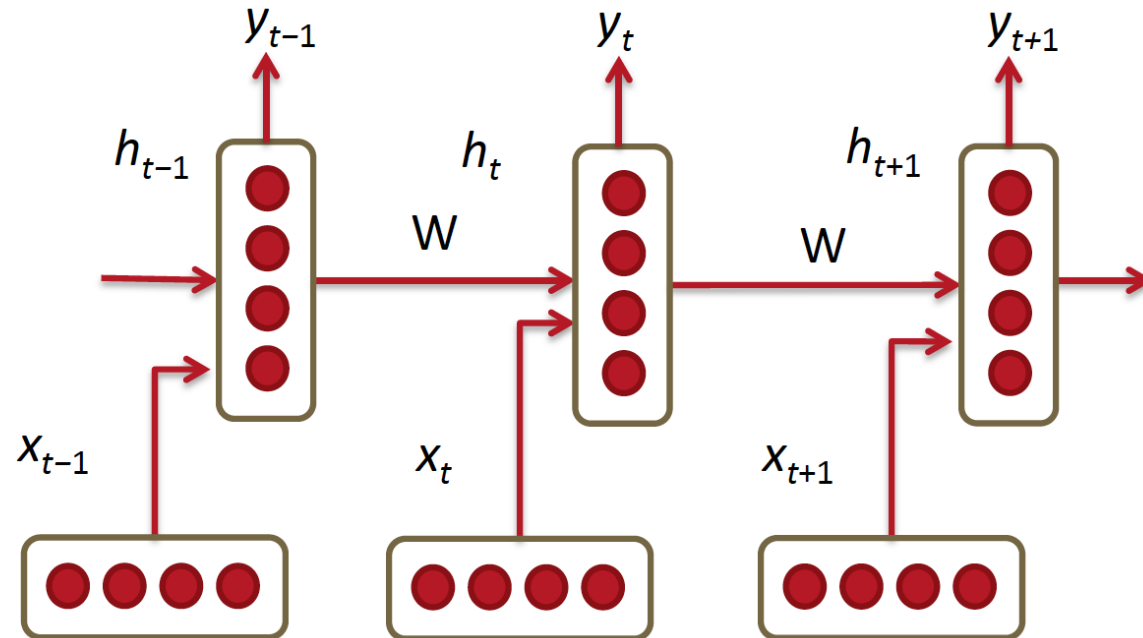
$$h_t = \sigma \left(W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]} \right)$$

$$\hat{y}_t = \text{softmax} \left(W^{(S)} h_t \right)$$



(Discrete-time) RNN

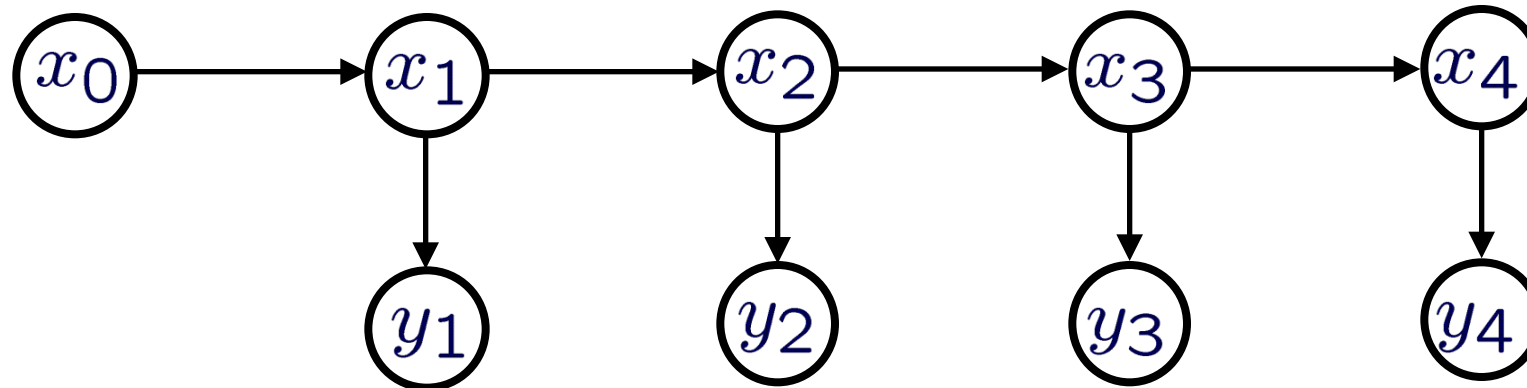
- Three time steps and beyond



- **A layered feedforward net**
- **Tied weights** for different time steps
- **Conditioning** (memorizing?) on all previous input
- Cheap to save memory in RAM

Detour: Hidden Markov Model

- A probabilistic model of sequences



- Emission probability:

$$P(y_i | x_i)$$

- Transition probability:

$$P(x_i | x_{i-1})$$

- Initial probability:

$$P(x_0)$$

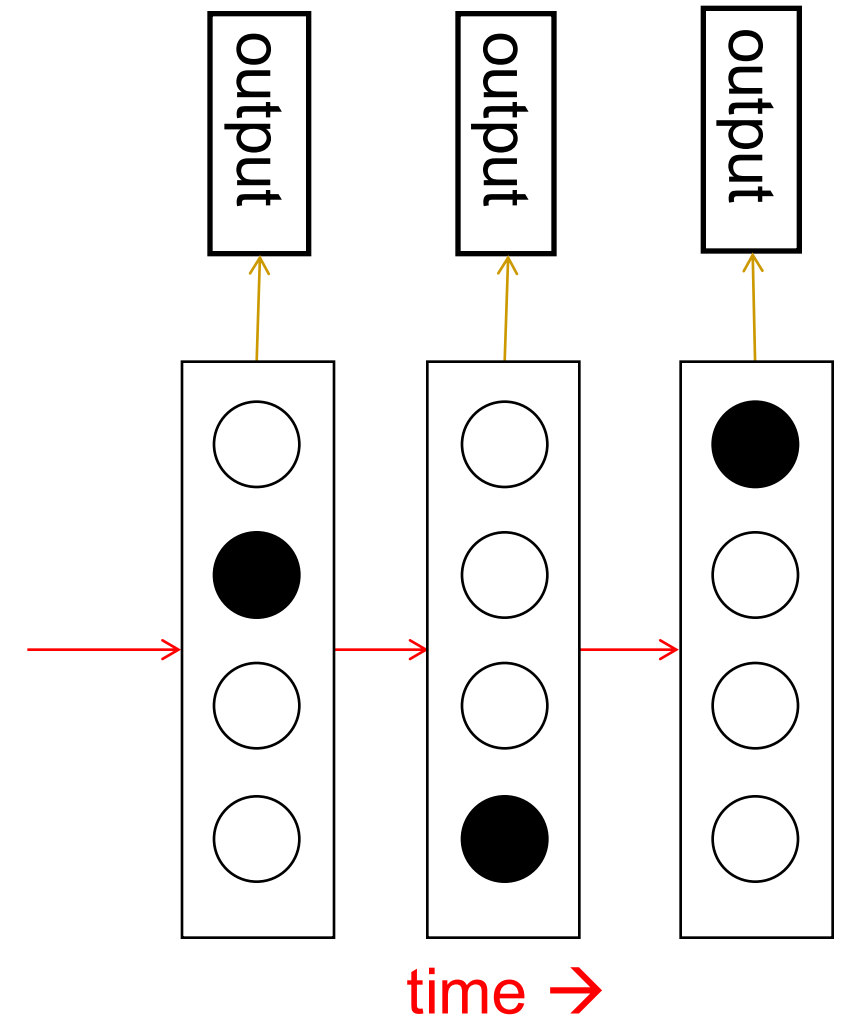
Detour: Hidden Markov Model

- Useful for modeling sequences
- Discrete hidden states, which satisfy Markov assumption
- Inference and learning (optional)
 - Evaluation: forward probability
 - Decoding: forward-backward algorithm, Viterbi decoding
 - Learning: EM algorithm (Baum-Welch)

Begin: Slides from Geoffrey Hinton

Hidden Markov Models (computer scientists love them!)

- Hidden Markov Models have a discrete one-of-N hidden state. Transitions between states are stochastic and controlled by a transition matrix. The outputs produced by a state are stochastic.
 - We cannot be sure which state produced a given output. So the state is “hidden”.
 - It is easy to represent a probability distribution across N states with N numbers.
- To predict the next output we need to infer the probability distribution over hidden states.
 - HMMs have efficient algorithms for inference and learning.

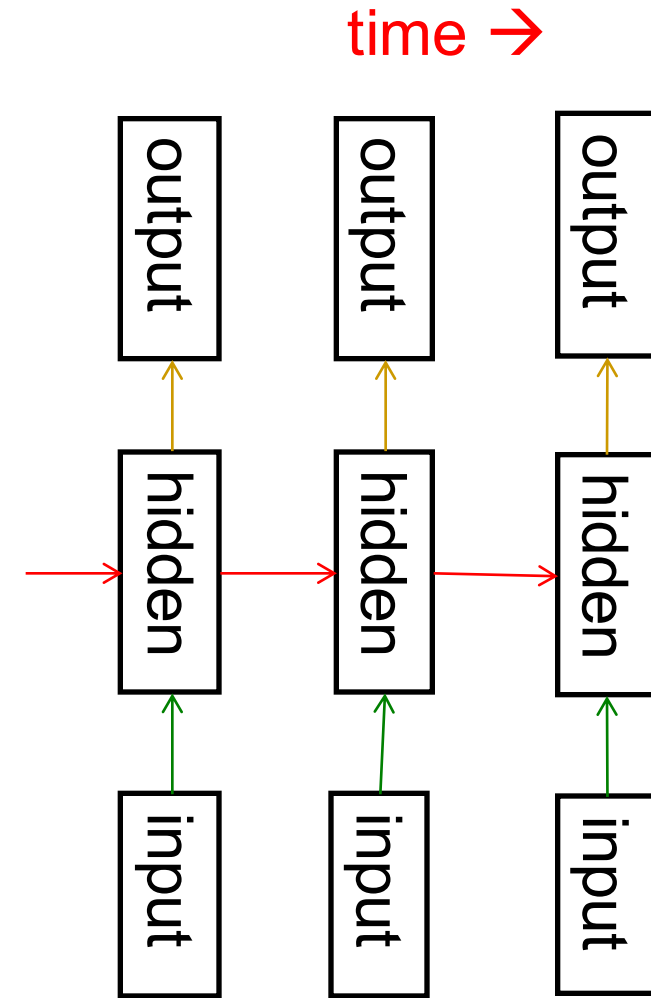


A fundamental limitation of HMMs

- Consider what happens when a hidden Markov model generates data.
 - At each time step it must select one of its hidden states. So with N hidden states it can only remember $\log(N)$ bits about what it generated so far.
- Consider the information that the first half of an utterance contains about the second half:
 - The syntax needs to fit (e.g. number and tense agreement).
 - The semantics needs to fit. The intonation needs to fit.
 - The accent, rate, volume, and vocal tract characteristics must all fit.
- All these aspects combined could be 100 bits of information that the first half of an utterance needs to convey to the second half. 2^{100} is big!

Recurrent neural networks

- RNNs are very powerful, because they combine two properties:
 - Distributed hidden state that allows them to store a lot of information about the past efficiently.
 - Non-linear dynamics that allows them to update their hidden state in complicated ways.
- With enough neurons and time, RNNs can compute anything that can be computed by your computer.



Do generative models need to be stochastic?

- Linear dynamical systems and hidden Markov models are stochastic models.
 - But the posterior probability distribution over their hidden states given the observed data so far is a deterministic function of the data.
- Recurrent neural networks are deterministic.
 - So think of the hidden state of an RNN as the equivalent of the deterministic probability distribution over hidden states in a linear dynamical system or hidden Markov model.

Recurrent neural networks

- What kinds of behaviour can RNNs exhibit?
 - They can oscillate. [Good for motor control?](#)
 - They can settle to point attractors. [Good for retrieving memories?](#)
 - They can behave chaotically. [Bad for information processing?](#)
 - RNNs could potentially learn to implement lots of small programs that each capture a nugget of knowledge and run in parallel, interacting to produce very complicated effects.
- But the computational power of RNNs makes them very hard to train.
 - For many years we could not exploit the computational power of RNNs despite some heroic efforts (e.g. Tony Robinson's speech recognizer).

End: Slides from Geoffrey Hinton

Today

- Administrivia
- Recurrent Neural Networks (RNNs) (II)
- OCR in the wild, by Aisha

Upload slides before or after class

- See “Paper Presentation” on UCF webcourse
- Sharing your slides
 - **Refer to the original sources of images, figures, etc. in your slides**
 - Convert them to a PDF file
 - Upload the PDF file to “Paper Presentation” after your presentation

Reading Text in the Wild with Convolutional Neural Networks

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, Andrew
Zisserman, IJCV 2016
Visual Geometry Group
University of Oxford

Aisha Urooj

Paper's Contribution

- A novel text recognition method—A deep convolutional neural network (CNN) which takes the whole word image as the input
 - Model is trained purely on synthetic data without any human labeling
- A novel detection strategy for text spotting: the use of fast region proposal methods to perform word detection.
- The application of pipeline for large-scale visual search of text in video

Outline

- Motivation
- Approach overview
- Details of pipeline stages
- Experimentation results
- Conclusion

Motivation

Visual Understanding:

Important challenge:

Text spotting: Automatic detection and recognition of text in natural images

- Can be used to decode and use semantic content of visual media for understanding , annotating and retrieving the billions of images produced on the daily basis.

Motivation (2)

- Traditionally, text recognition has been focused on document images.
 - OCR techniques are well suited to digitize planar, paper-based documents.
- OCR techniques fail on natural scene images
 - OCR approaches are tuned to the largely black-and-white, line based printed documents.

Motivation (3)

- Challenges for text that occurs in natural scenes:
 - huge variance in appearance and layout,
 - Large number of fonts and styles
 - Inconsistent lighting
 - Occlusions
 - Orientations
 - Noise
 - Presence of background objects causing false-positive detections

Motivation (4)

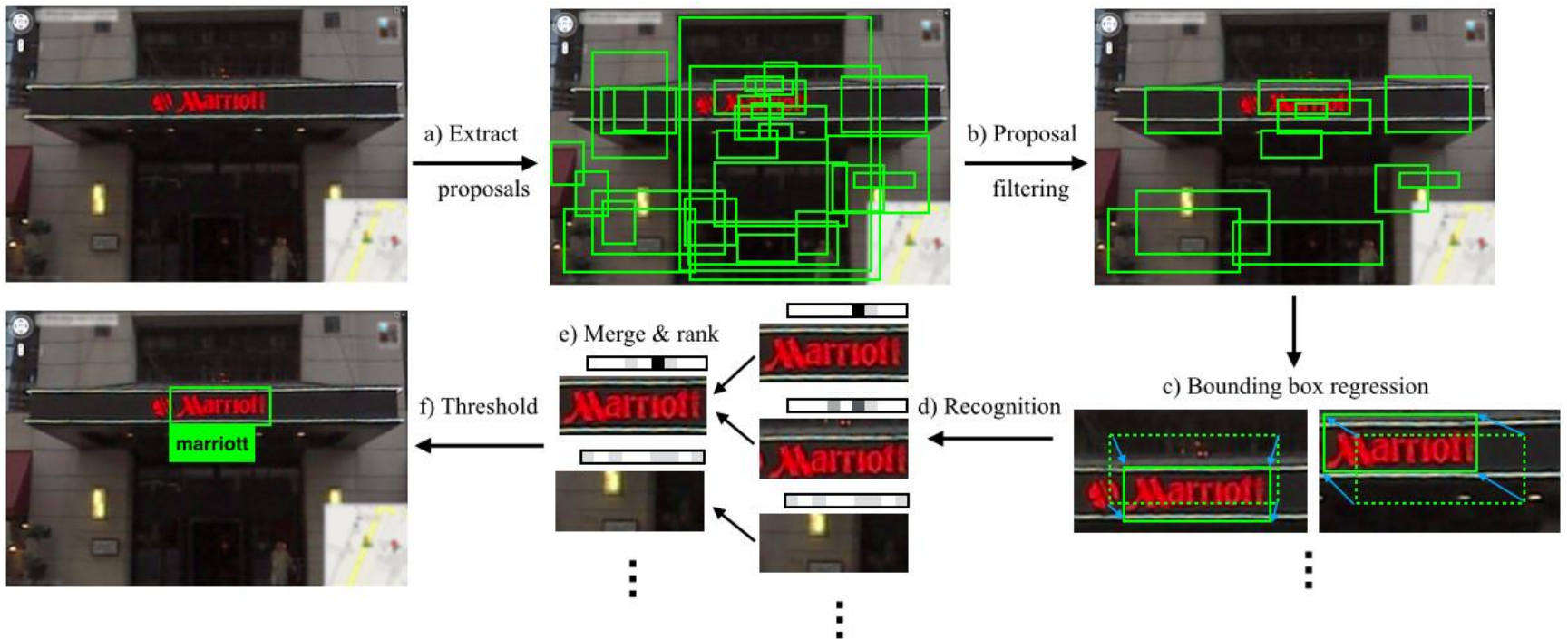


Feature Standard Deviation Analysis of Structure Elements (FSDASE)

The goal of the FSDASE is to find those DSEs that have maximum SD at the text blocks and minimum SD at the non-text blocks and the opposite.

- A training dataset is required
- Does not cause a problem because such dataset already is required for the training of the SVMs
- Therefore the final block descriptor is a vector with 32 elements and it corresponds to the frequency of the 32 DSEs that the block contains

Overview of complete pipeline



1. Proposal Generation

- Edge Box
- Aggregate Channel Features Detector

Edge Boxes

Key Intuition:

- Objects are generally self contained.
- The number of contours wholly enclosed by a bounding box shows likelihood of the box containing an object.
- Edges crossing a boundary box suggest there is an object not wholly contained by the bounding box.
- Words: Collection of characters with sharp boundaries
 - Object here would be collection of boundaries

Edge Boxes

- The edge response map is computed using the Structured Edge detector
- Non-Maximal Suppression is used orthogonal to the edge responses, sparsifying the edge map
- A score s_b based on the number of edges wholly contained by candidate bounding box b is computed
- The boxes b are evaluated in a sliding window manner, over multiple scales and aspect ratios, and given a score s_b .
- The boxes are sorted by score and non-maximal suppression is performed
- A box is removed if its overlap with another box of higher score is more than a threshold
- Output: A set B_e of candidate bounding boxes.

Aggregate Channel Feature Detector

- A conventional sliding window detector based on ACF features coupled with an AdaBoost classifier
- Used for its computational speed
- For each image I a number of feature channels are computed, such that channel $C = \Omega(I)$
 - Ω is the channel feature extraction function.
- Channels:
 - normalised gradient magnitude,
 - histogram of oriented gradients (6 channels),
 - and the raw greyscale input

Aggregate Channel Feature Detector

- **Aggregate channels features(ACF):** Each channel C is smoothed, divided into blocks and the pixels in each block are summed and smoothed again.
- ACF features are not scale-invariant.
 - For multiscale detection, extract features at many different scales.
- In a standard detection pipeline, the channel features for a particular scale s are computed by resampling the image and recomputing the channel features

$$C_s = \Omega(I_s) \quad I_s = R(I, s)$$

Aggregate Channel Feature Detector

- Computationally expensive
- The channel features at scale s can be approximated by resampling the features at a different scale

$$C_s \approx R(C, s) \cdot s^{-\lambda\Omega},$$

- Fast feature pyramids can be computed by evaluating $C_s = \Omega(R(I, s))$ at only a single scale per octave:

$$(s \in \{1, \frac{1}{2}, \frac{1}{4}, \dots\})$$

$$C_s = R(C_{s'}, s/s')(s/s')^{-\lambda\Omega} \quad s' \in \{1, \frac{1}{2}, \frac{1}{4}, \dots\}$$

Aggregate Channel Feature Detector

- Classifier was evaluated on every block of aggregate channel features in feature pyramid, for multiple aspect ratios (for different word lengths) giving score for each box.
- Thresholding on score gives a set of word proposal bounding boxes from the detector, B_d

Proposal Generation (contd..)

- The final set of candidate bounding boxes:

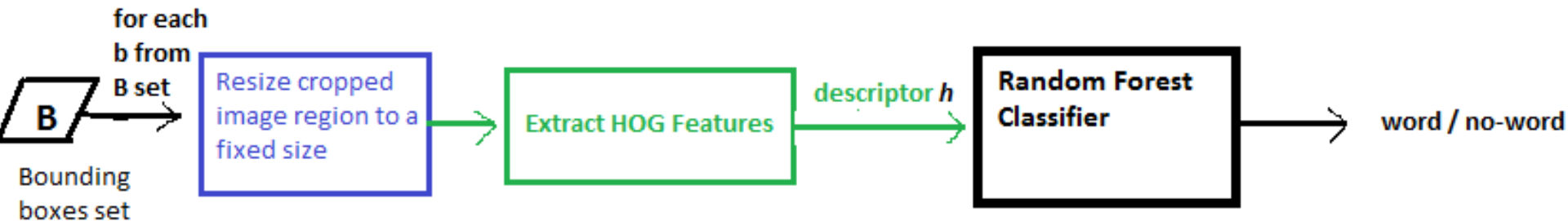
$$B = \{B_e \cup B_d\}$$

- The Edge Boxes and the ACF detector do not achieve particularly higher recall when used independently.
 - 92% and 70% respectively
 - But when proposals are combined, they achieve 98% recall (0.5 overlap threshold)

2. Filtering & Refinement

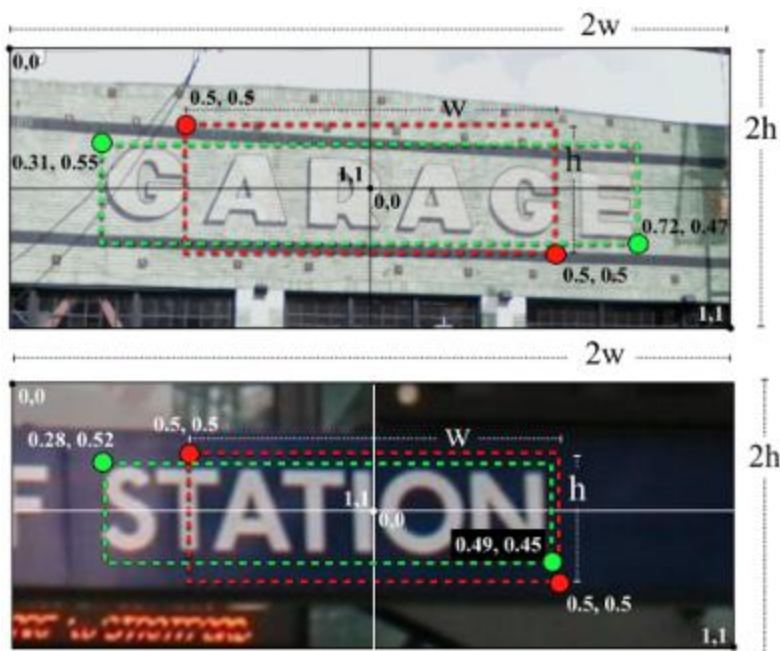
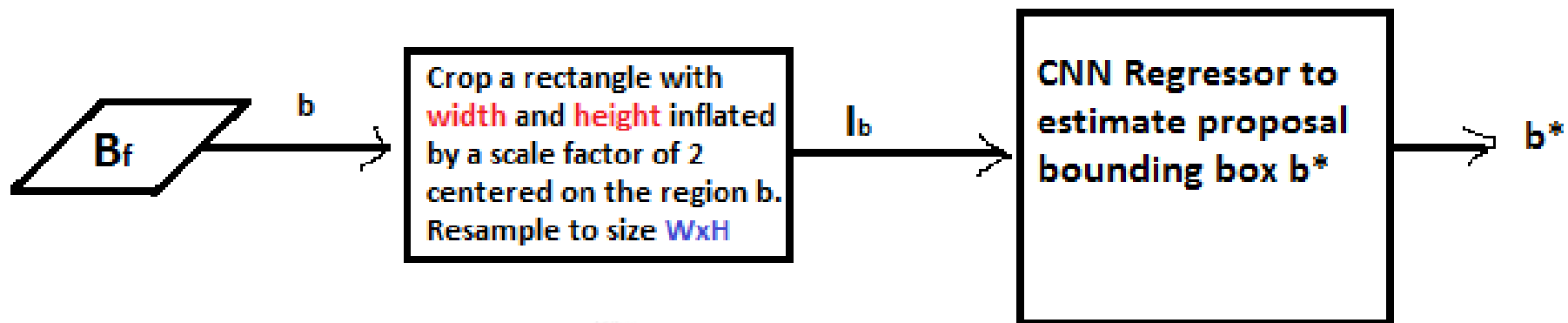
- Thousands of bounding boxes are generated to achieve high recall
 - Most of them are false-positives
 - Need to filter these out to a computationally manageable number
- Train the regressor to refine the location of the bounding boxes
 - Word Classification
 - Bounding Box Regression

Word Classification



- Output: Filtered set of bounding boxes B_f

Bounding Box Regression (1)



Red: Original proposal
Green: Adjusted proposal

The cropped input image shown is always centred on the original proposal, meaning the original proposal always has implied encoded coordinates of (0.5,0.5,0.5,0.5)

Bounding Box Regression (2)

$$\min_{\Phi} \sum_{b \in B_{train}} \|g(I_b; \Phi) - q(b_{gt})\|_2^2$$

Where,

- B_{train} is training set
- Φ : network parameters
- G : CNN forward pass function
- q : bounding box coordinate encoder

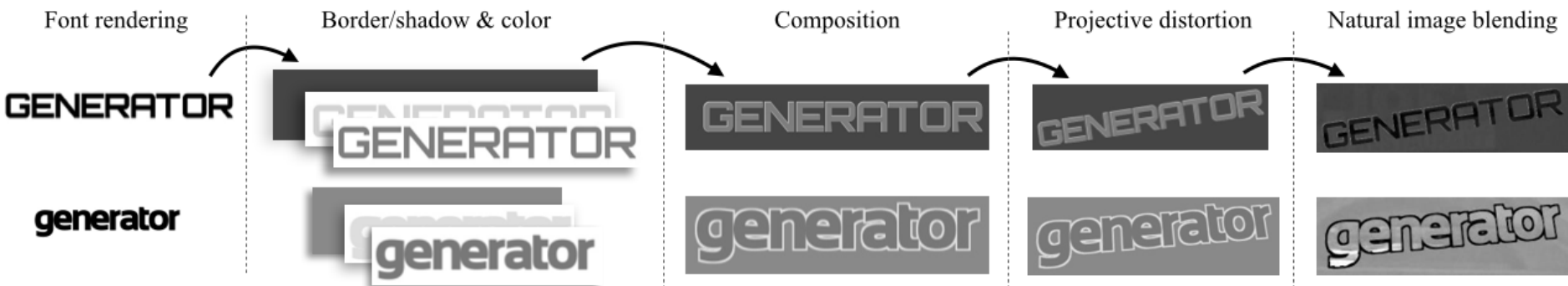
3. Text Recognition

- A deep CNN is used to perform classification across a pre-defined dictionary of words
- The model, can scale to a huge dictionary of 90k words
 - encompasses the majority of the commonly used English language.
- Training data requirement: Many training samples of every different possible word must be gathered
- Such a training dataset does not exist
 - Used synthetic training data to train CNN

Text Recognition – Synthetic Data Generation Process

- Font rendering
- Border/Shadow rendering
- Base coloring
- Projective distortion
- Natural data blending
- Noise

Text Recognition – Synthetic Training Data



- This dataset consists of **9 million images** covering **90k English words**, and includes the training, validation and test splits used in our work.



Some randomly sampled data created by the synthetic text engine

CNN Model

- Multiclass classification
- One class per word
- Words w are constrained to be selected in a pre-defined dictionary W
- 5 convolutional layers, 3 fully connected layers
- Predicted word recognition result w^* out of the set of all dictionary words W in a language L for a given input image x is:

$$w^* = \arg \max_{w \in W} P(w|x, \mathcal{L}).$$

CNN Model

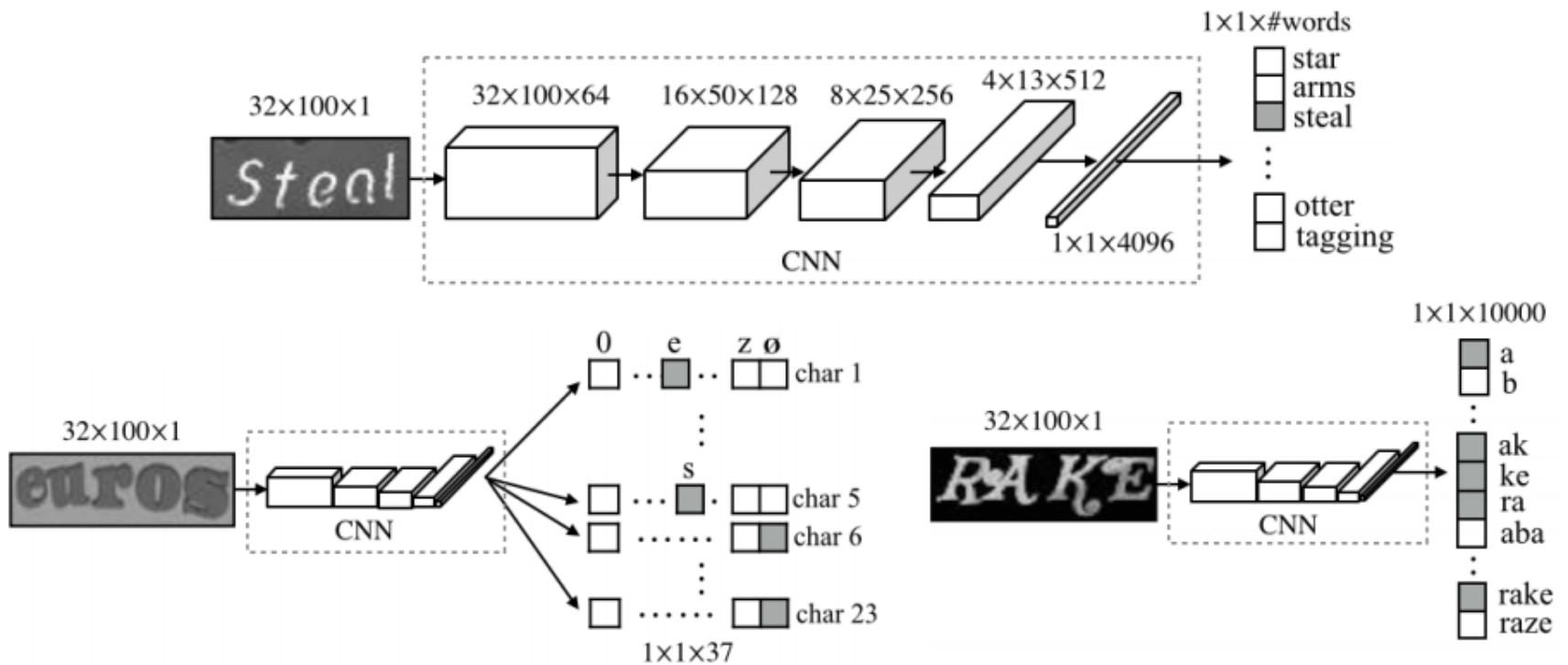
$$P(w|x, \mathcal{L}) = \frac{P(w|x)P(w|\mathcal{L})P(x)}{P(x|\mathcal{L})P(w)}$$

Assuming that x is independent of \mathcal{L} and that prior to any knowledge of language all words are equally probable, scoring function reduces to :

$$w^* = \arg \max_{w \in \mathcal{W}} P(w|x)P(w|\mathcal{L}).$$

- The per-word output probability $P(w|x)$ is modelled by the softmax output of the final fully-connected layer of the recognition CNN
- The language based word prior $P(w|\mathcal{L})$ can be modelled by a lexicon or frequency counts

CNN Model



A schematic of the CNN used for text recognition by word classification. The dimensions of the feature maps at each layer of the network are shown

4.Merging and Ranking

- This set of detections still contains a number of false-positive and duplicate detections of words
- A final merging and ranking of detections must be performed depending on the task at hand: text spotting or text based image retrieval

Merging and Ranking – Text Spotting

- **Goal:** Each word should be labelled by a bounding box enclosing the word and the bounding box should have an associated text label.
- For this, assign each bounding box a label w_b and score s_b according to b 's maximum word probability

$$w_b = \arg \max_{w \in \mathcal{W}} P(w|b, I), \quad s_b = \max_{w \in \mathcal{W}} P(w|b, I)$$

Merging and Ranking – Text Spotting

- Cluster duplicate detections of the same word instance:
 - Performed a greedy non maximum suppression (NMS) on detections with the same word label
 - Aggregated the scores of suppressed proposals.
- To improve the overlap of detection results, multiple rounds of bounding box regression and NMS were additionally performed to further refine detections.
 - A Recurrent regressor network



An example of the improvement in localisation of the word detection `pharmacy` through multiple rounds of recurrent regression

Merging and Ranking – Image Retrieval

- Retrieve the list of images which contain the given query words.
- Localisation of the query word is optional.
- At query time, assign each image a score for the query words $Q = \{q_1, q_2, \dots\}$
- Sorting the images in the database I in descending order of score.
- The per-image probability distribution across word space $P(w|I)$ by averaging the word probability distributions across all detections B_f in an image

$$p_I = P(w|I) = \frac{1}{|B_f|} \sum_{b \in B_f} p_b.$$

Merging and Ranking – Image Retrieval

- This distribution is computed offline for all images in database.
- At query time, simply compute a score for each image representing the probability that the image I contains any of the query words Q .

$$s_I^Q = \sum_{q \in Q} P(q|I) = \sum_{q \in Q} p_I(q)$$

Experiments - Datasets

Table 1 A description of the various *text recognition* datasets evaluated on

Label	Description	Lex. size	# Images
Synth	Our synthetically generated test dataset	90k	900k
IC03	ICDAR 2003 (http://algoval.essex.ac.uk/icdar/datasets.html) test dataset	–	860
IC03-50	ICDAR 2003 (http://algoval.essex.ac.uk/icdar/datasets.html) test dataset with fixed lexicon	50	860
IC03-Full	ICDAR 2003 (http://algoval.essex.ac.uk/icdar/datasets.html) test dataset with fixed lexicon	563	860
SVT	SVT (Wang et al. 2010) test dataset	–	647
SVT-50	SVT (Wang et al. 2010) test dataset with fixed lexicon	50	647
IC13	ICDAR 2013 (Karatzas et al. 2013) test dataset	–	1015
IIIT5k-50	IIIT5k (Mishra et al. 2012) test dataset with fixed lexicon	50	3000
IIIT5k-1k	IIIT5k (Mishra et al. 2012) test dataset with fixed lexicon	1000	3000

Experiments - Datasets

Table 2 A description of the various *text spotting* datasets evaluated on

Label	Description	Lex. size	# Images
IC03	ICDAR 2003 (http://algoval.essex.ac.uk/icdar/datasets.html) test dataset	–	251
IC03-50	ICDAR 2003 (http://algoval.essex.ac.uk/icdar/datasets.html) test dataset with fixed lexicon	50	251
IC03-Full	ICDAR 2003 (http://algoval.essex.ac.uk/icdar/datasets.html) test dataset with fixed lexicon	860	251
SVT	SVT (Wang et al. 2010) test dataset	–	249
SVT-50	SVT (Wang et al. 2010) test dataset with fixed lexicon	50	249
IC11	ICDAR 2011 (Shahab et al. 2011) test dataset	–	255
IC13	ICDAR 2013 (Karatzas et al. 2013) test dataset	–	233

Experiments - Datasets

Table 3 A description of the various *text retrieval* datasets evaluated on

Label	Description	# queries	# Images
IC11	ICDAR 2011 (Shahab et al. 2011) test dataset	538	255
SVT	SVT (Wang et al. 2010) test dataset	427	249
STR	IIIT STR (Mishra et al. 2013) text retrieval dataset	50	10k
Sports	IIIT Sports-10k (Mishra et al. 2013) text retrieval dataset	10	10k
BBC News	A dataset of keyframes from BBC News video	–	2.3 m

Results (1) – The recall and average no. of proposals.

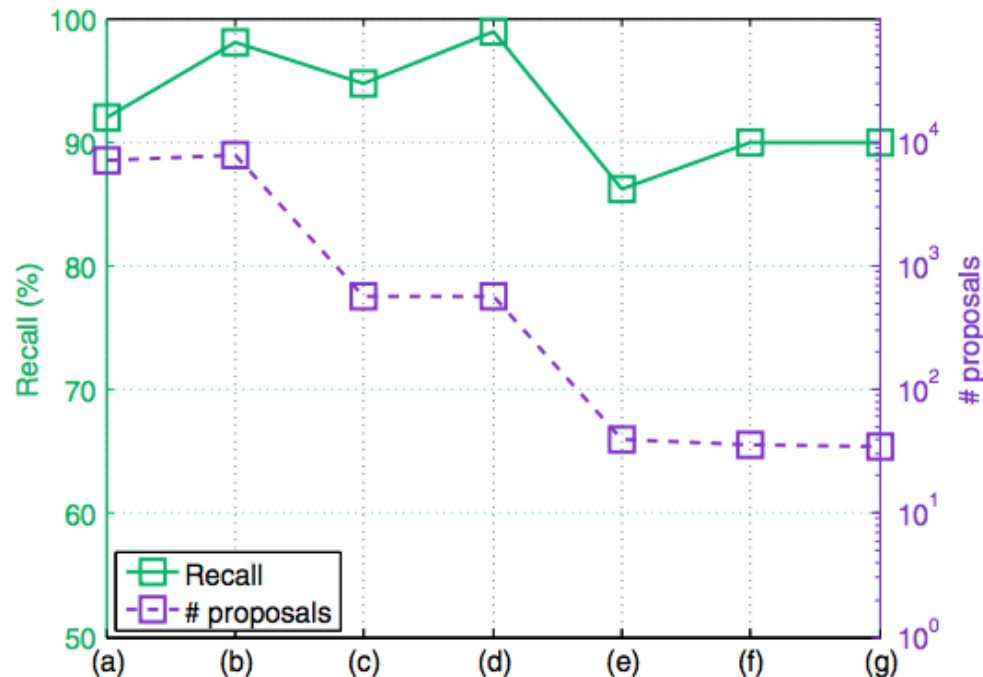


Fig. 7 The recall and the average number of proposals per image on IC03 after each stage of the pipeline: (a) Edge Box proposals, (b) ACF detector proposals, (c) proposal filtering, (d) bounding box regression, (e) regression NMS round 1, (f) regression NMS round 2, (g) regression NMS round 3. The recall computed is detection recall across the dataset (*i.e.* ignoring the recognition label) at 0.5 overlap. The detection precision is 13% at the end of the pipeline (g)

Results (2) – Overlap Recall of different Region Proposal algorithms

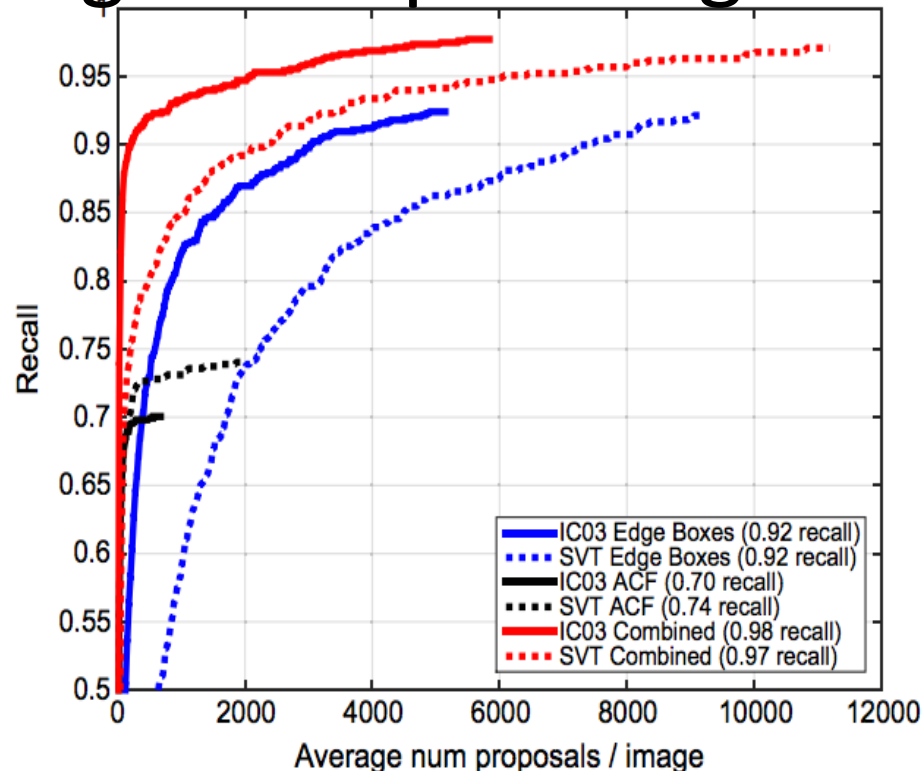


Fig. 8 The 0.5 overlap recall of different region proposal algorithms. The recall displayed in the legend for each method gives the maximum recall achieved. The curves are generated by decreasing the minimum score for a proposal to be valid, and terminate when no more proposals can be found. Due to the large number of region proposals and the small number of words contained in each image the precision is negligible to achieve high levels of recall (Color figure online)

Results (3) – Recognition Accuracies

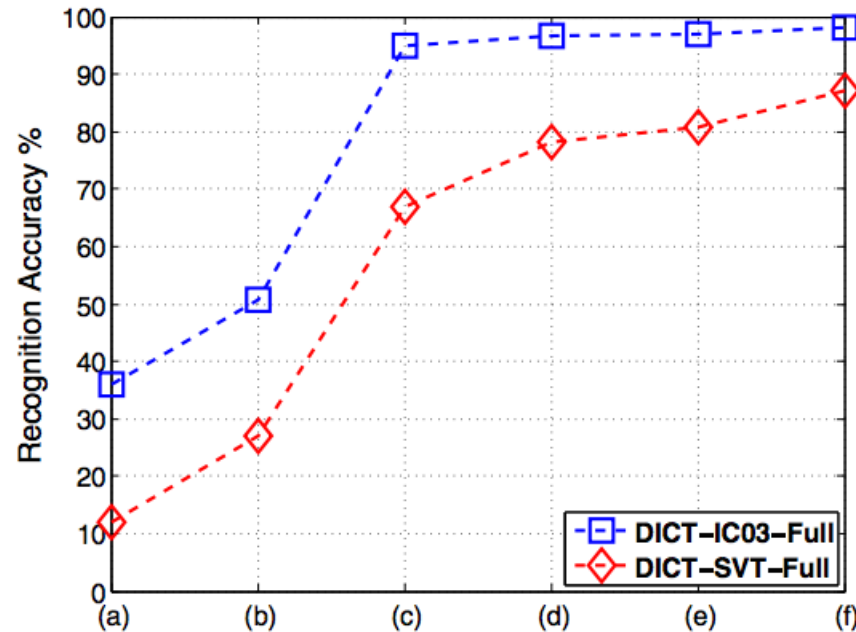


Fig. 9 The recognition accuracies of text recognition models trained on just the IC03 lexicon (DICT-03-Full) and just the SVT lexicon (DICT-SVT-Full), evaluated on IC03 and SVT respectively. The models are trained on purely synthetic data with increasing levels of sophistication of the synthetic data. (a) Black text rendered on a white background with a single font, Droid Sans. (b) Incorporating all of Google fonts. (c) Adding background, foreground, and border colouring. (d) Adding perspective distortions. (e) Adding noise, blur and elastic distortions. (f) Adding natural image blending—this gives an additional 6.2% accuracy on SVT. The final accuracies on IC03 and SVT are 98.1 and 87.0% respectively

Results (4) – Recognition Accuracies

Table 4 Comparison to previous methods for text recognition accuracy—where the groundtruth cropped word image is given as input

Model	Cropped word recognition accuracy (%)								
	Synth	IC03-50	IC03-Full	IC03	SVT-50	SVT	IC13	IIIT5k-50	IIIT5k-1k
<i>Baseline (ABBYY) (Wang et al. 2011; Yao et al. 2014)</i>	–	56.0	55.0	–	35.0	–	–	24.3	–
Wang et al. (2011)	–	76.0	62.0	–	57.0	–	–	–	–
Mishra et al. (2012)	–	81.8	67.8	–	73.2	–	–	64.1	57.5
Novikova et al. (2012)	–	82.8	–	–	72.9	–	–	–	–
Wang et al. (2012)	–	90.0	84.0	–	70.0	–	–	–	–
Goel et al. (2013)	–	89.7	–	–	77.3	–	–	–	–
PhotoOCR Bissacco et al. (2013)	–	–	–	–	90.4	78.0	87.6	–	–
Alsharif and Pineau (2014)	–	93.1	88.6	85.1 ^a	74.3	–	–	–	–
Almazán et al. (2014)	–	–	–	–	89.2	–	–	91.2	82.1
Yao et al. (2014)	–	88.5	80.3	–	75.9	–	–	80.2	69.3
Jaderberg et al. (2014)	–	96.2	91.5	–	86.1	–	–	–	–
Gordo (2014)	–	–	–	–	90.7	–	–	93.3	86.6
Proposed	95.2	98.7	98.6	93.1	95.4	80.7	90.8	97.1	92.7

The ICDAR 2013 results given are case-insensitive. Bold results outperform previous state-of-the-art methods. The baseline method is from a commercially available document OCR system.

^a Recognition is constrained to a dictionary of 50k words

Results (5) – End-to-end text spotting

Table 5 Comparison to previous methods for end-to-end text spotting

Model	End-to-end text spotting (F-measure %)								
	IC03-50	IC03-Full	IC03	IC03 ^b	SVT-50	SVT	IC11	IC11 ^b	IC13
Neumann and Matas (2011)	–	–	–	41	–	–	–	–	–
Wang et al. (2011)	68	51	–	–	38	–	–	–	–
Wang et al. (2012)	72	67	–	–	46	–	–	–	–
Neumann and Matas (2013)	–	–	–	–	–	–	–	45	–
Alsharif and Pineau (2014)	77	70	63 ^a	–	48	–	–	–	–
Jaderberg et al. (2014)	80	75	–	–	56	–	–	–	–
Proposed	90	86	78	72	76	53	76	69	76
Proposed (0.3 IoU)	91	87	79	73	82	57	77	70	77

Bold results outperform previous state-of-the-art methods

^a Recognition is constrained to a dictionary of 50k words

^b Evaluation protocol described in Neumann and Matas (2013)

Results (6) – Precision/Recall curves on different datasets

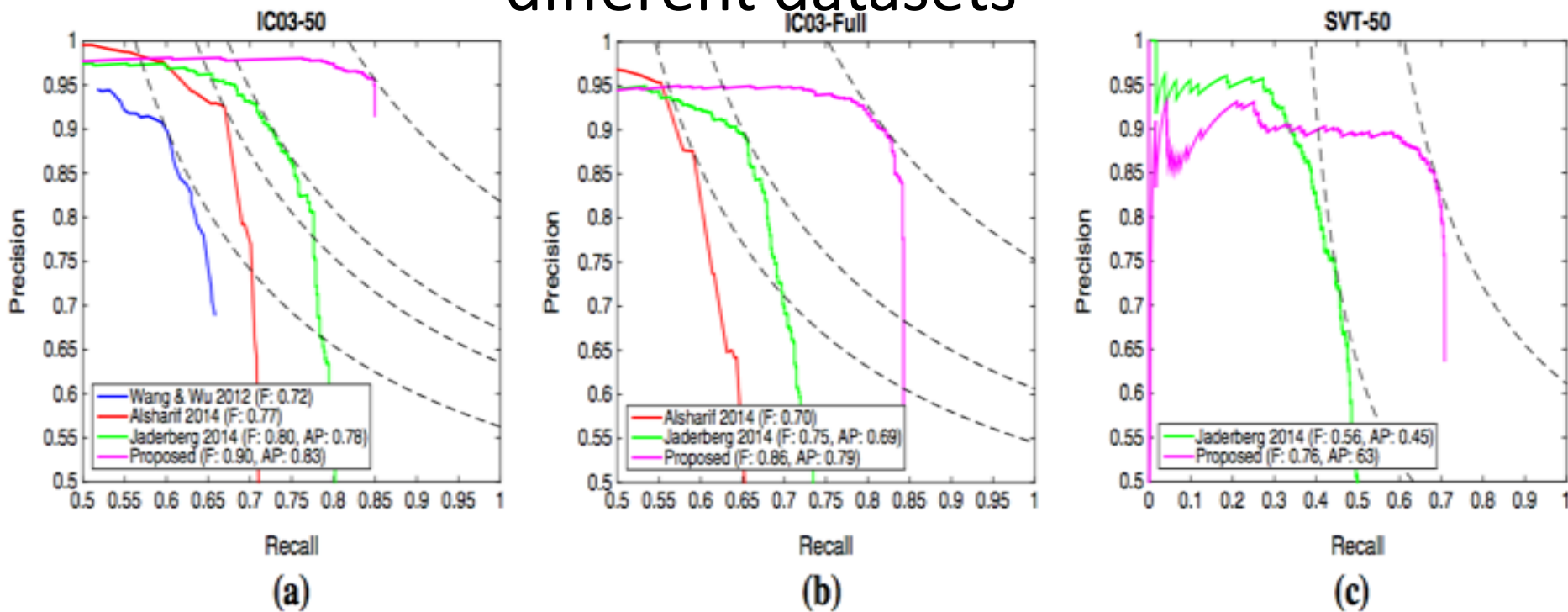


Fig. 10 The precision/recall curves on **a** the IC03-50 dataset, **b** the IC03-Full dataset, and **c** the SVT-50 dataset. The lines of constant F-measure are shown at the maximum F-measure point of each curve.

The results from [Alsharif and Pineau \(2014\)](#), [Wang et al. \(2012\)](#) were extracted from the papers (Color figure online)

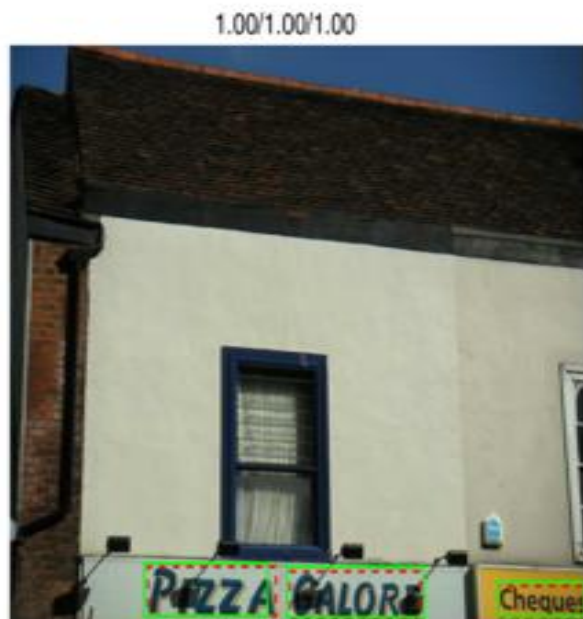


Fig. 11 Some example text spotting results from SVT-50 (top row) and IC11 (bottom row). Red dashed shows groundtruth and green shows correctly localised and recognised results. P/R/F figures are given above each image (Color figure online)

Results (7) – Text based Image Retrieval

Table 6 Comparison to previous methods for text based image retrieval

Model	IC11 (mAP)	SVT (mAP)	STR (mAP)	Sports (mAP)	Sports (P@10)	Sports (P@20)
Wang et al. (2011) ^a	–	21.3	–	–	–	–
Neumann and Matas (2012) ^a	–	23.3	–	–	–	–
Mishra et al. (2013)	65.3	56.2	42.7	–	44.8	43.4
Proposed	90.3	86.3	66.5	66.1	91.0	92.5

We report mean average precision (mAP) for IC11, SVT, STR, and Sports, and also report top- n retrieval to compute precision at n (P@ n) on Sports. Bold results outperform previous state-of-the-art methods

^a Experiments were performed by [Mishra et al. \(2013\)](#), not by the original authors

Results (8) – Text based Image Retrieval

Fig. 12 An illustration of the problems with incomplete annotations in test datasets. We show examples of the top two results for the query `apartments` on the SVT dataset and the query `castrol` on the Sports dataset. All retrieved images contain the query word (*green box*), but some of the results have incomplete annotations, and so although the query word is present the result is labelled as incorrect. This leads to a reported AP of 0.5 instead of 1.0 for the SVT query, and a reported P@2 of 0.5 instead of the true P@2 of 1.0 (Color figure online)

SVT: apartments
1. Score: 0.219 – Groundtruth: 0



2. Score: 0.022 – Groundtruth: 1



Sports: castrol
1. Score: 1.0 – Groundtruth: 1



2. Score: 1.0 – Groundtruth: 0



Table 7 The processing time for each stage of the pipeline evaluated on the SVT dataset on a single CPU core and single GPU

Stage	# Proposals	Time (s)	Time/proposal (ms)
(a) Edge Boxes	$>10^7$	2.2	<0.002
(b) ACF detector	$>10^7$	2.1	<0.002
(c) RF filter	10^4	1.8	0.18
(d) CNN regression	10^3	1.2	1.2
(e) CNN recognition	10^3	2.2	2.2

As the pipeline progresses from (a) to (e), the number of proposals is reduced (starting from all possible bounding boxes), allowing us to increase our computational budget per proposal while keeping the overall processing time for each stage comparable

hollywood - P@100: 100%



HOLLYWOOD



HOLLYWOOD

boris johnson - P@100: 100%



Boris Johnson



BORIS JOHNSON

vision - P@100: 93%



vision



VISION

Fig. 13 The top two retrieval results for three queries on our BBC News dataset—hollywood, boris johnson, and vision. The frames and associated videos are retrieved from 5k hours of BBC video.

We give the precision at 100 (P@100) for these queries, equivalent to the first page of results of our web application

Conclusion

- An end-to-end text reading pipeline—a detection and recognition system for text in natural scene images.
- Outperforms previous methods on text spotting and image retrieval tasks
- System is fast and scalable.
- The ability of recognition model to be trained purely on synthetic data allows system to be easily re-trained for recognition of other languages or scripts, without any human labelling effort.

Limitation

- All images are horizontally aligned.

