Sequence-to-sequence deep learning for eye movement classification

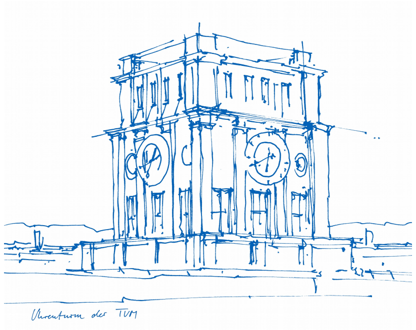
Mikhail Startsev

Ioannis Agtzidis

Michael Dorr

Technical University Munich

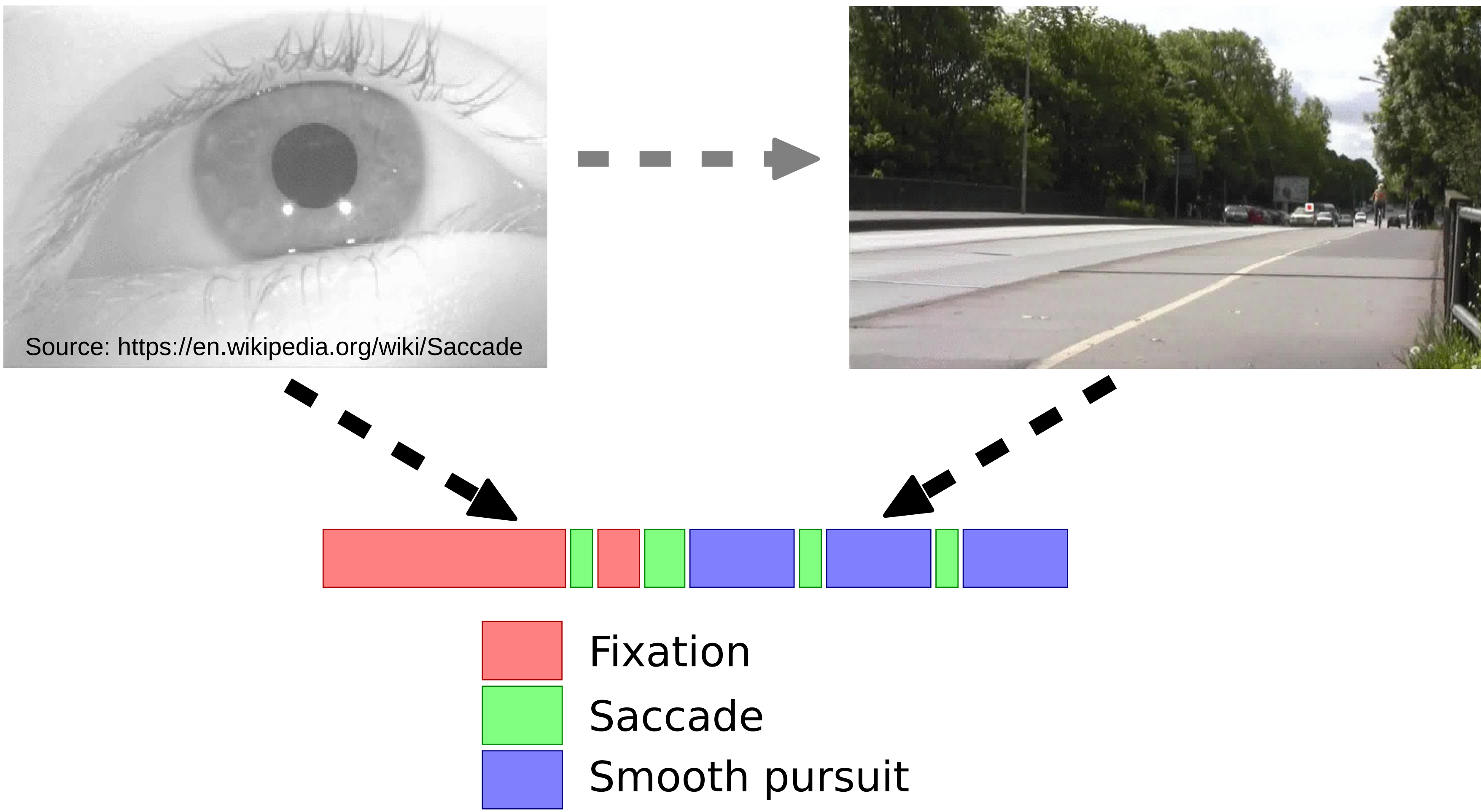
Institute for Human-Machine Communication



August 30, 2018

Trieste

What is eye movement classification?



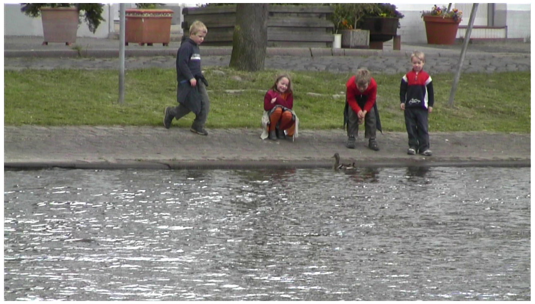
What is eye movement classification?

GazeCom data set (Dorr et al. 2010):

http://michaeldorr.de/smoothpursuit

- Dynamic natural scenes
- Eighteen 20-second clips
- 47 observers per clip (on average)
- Full manual annotation for major eye movement types (fixations, saccades, smooth pursuits, + “noise”)

→ over 4.5 hours of hand-labelled eye tracking recordings



Why do eye movement classification?

- Eye movement-level statistics and analysis
- High(er)-level analysis of eye tracking sessions (e.g. catch-up saccades during smooth pursuit)
- Benefits for eye movement-based interaction

Why do eye movement classification?

- Eye movement-level statistics and analysis
- High(er)-level analysis of eye tracking sessions (e.g. catch-up saccades during smooth pursuit)
- Benefits for eye movement-based interaction

Why not just use what is already built into the eye tracker software?

Why do eye movement classification?

- Eye movement-level statistics and analysis
- High(er)-level analysis of eye tracking sessions (e.g. catch-up saccades during smooth pursuit)
- Benefits for eye movement-based interaction

Why not just use what is already built into the eye tracker software?

- Simpler algorithms
- No smooth pursuit detection
  → poorer fixation detection in general
  → “elongated” “dynamic” fixations during dynamic stimuli viewing

How to do eye movement classification?

Classical algorithms

- Thresholding speed, dispersion, acceleration (I-VT, I-DT, I-VVT, I-VDT, Dorr et al. 2010)
- Hand-crafted features (I-VMP, Berg et al. 2009, Santini et al. 2012, Larsson et al. 2015)

Machine learning algorithms

- SVM (Anantrasirichai et al. 2016)
- Convolutional neural networks (Hoppe and Bulling 2016)
- Clustering (Agtzidis et al. 2016)
- Random forests (Zemblys et al. 2017)

How to do eye movement classification?

Classical algorithms

- Thresholding speed, dispersion, acceleration (I-VT, I-DT, I-VVT, I-VDT, Dorr et al. 2010)
- Hand-crafted features (I-VMP, Berg et al. 2009, Santini et al. 2012, Larsson et al. 2015)

Machine learning algorithms

- SVM (Anantrasirichai et al. 2016)
- Convolutional neural networks (Hoppe and Bulling 2016)
- Clustering (Agtzidis et al. 2016)
- Random forests (Zemblys et al. 2017)

Does not use sequential information

How to do eye movement classification?

Classical algorithms

- Thresholding speed, dispersion, acceleration (I-VT, I-DT, I-VVT, I-VDT, Dorr et al. 2010)
- Hand-crafted features (I-VMP, Berg et al. 2009, Santini et al. 2012, Larsson et al. 2015)

(essentially)
Sample-level classification

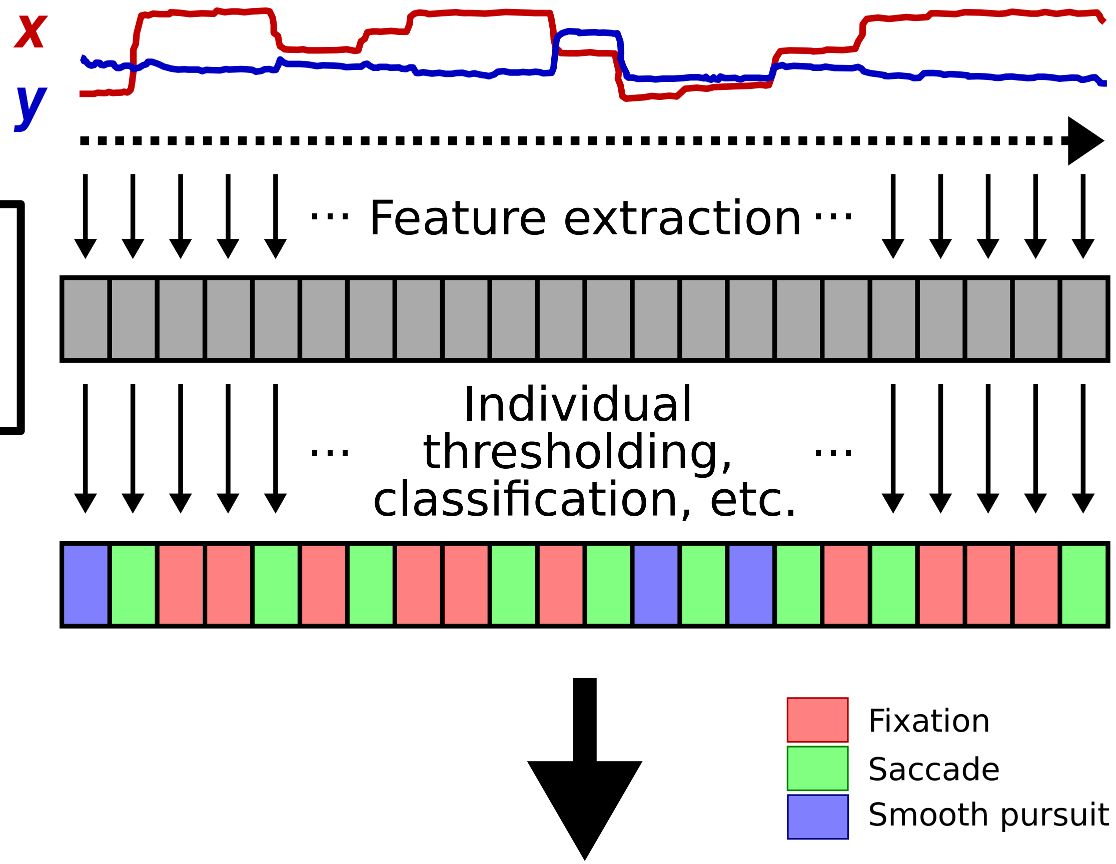
Machine learning algorithms

- SVM (Anantrasirichai et al. 2016)
- Convolutional neural networks (Hoppe and Bulling 2016)
- Clustering (Agtzidis et al. 2016)
- Random forests (Zemblys et al. 2017)

Does not use sequential information

How to do eye movement classification?

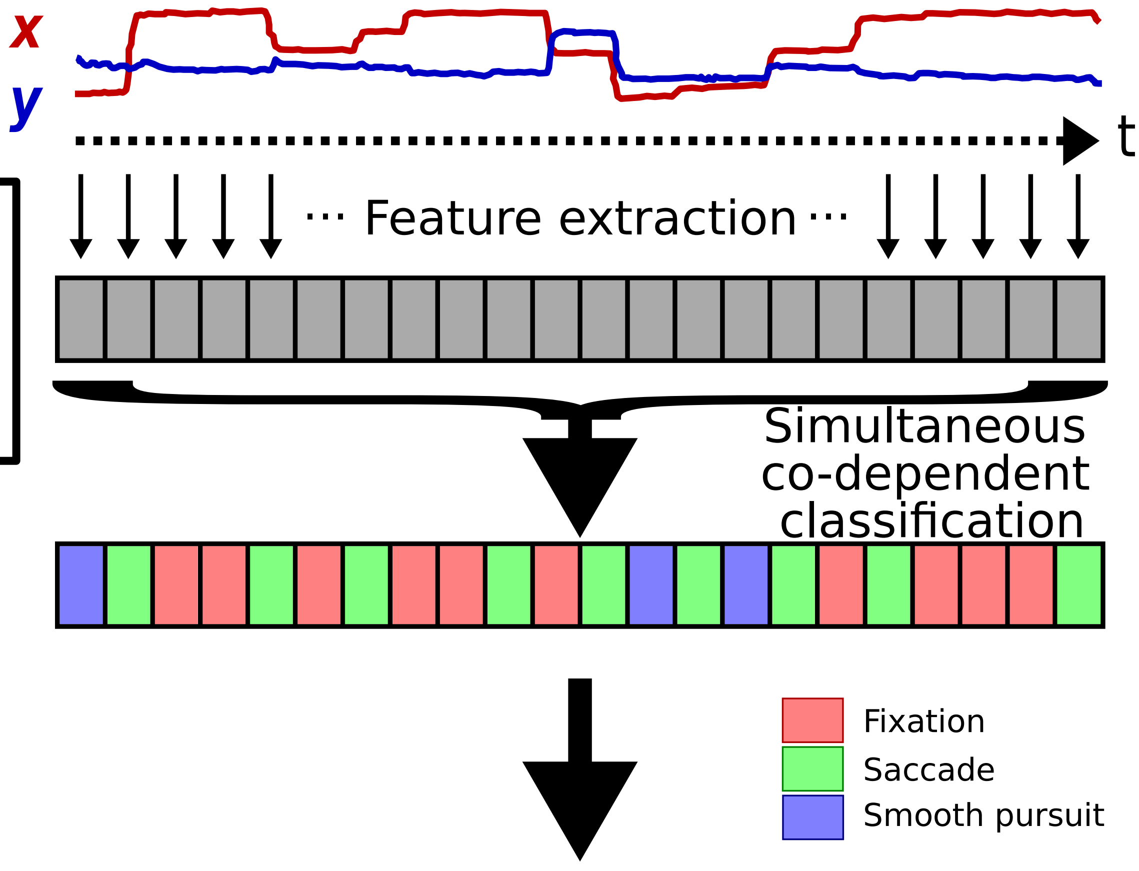
Sample-level classification



Evaluation
(per-class F1 scores for samples and for whole episodes)

How to do eye movement classification?

Sequence-to-sequence classification



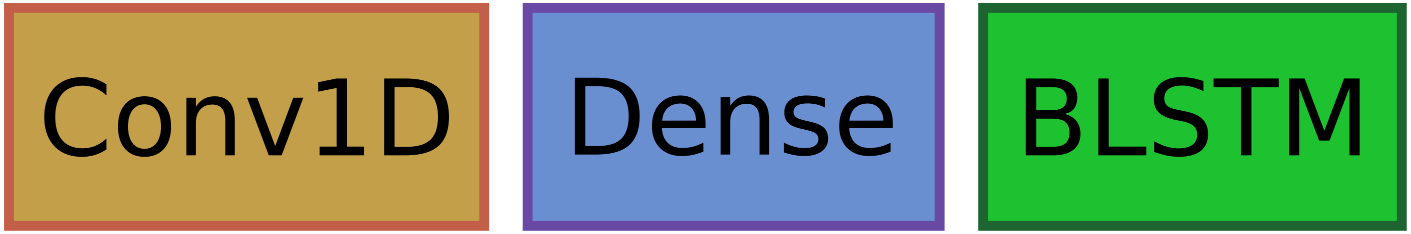
Evaluation
(per-class F1 scores for samples and for whole episodes)

Our model

A typical combination of layers for deep sequence-to-sequence processing:

- Convolutional (1D in our case)
- Dense
- Long short-term memory – LSTM (bidirectional in our case)

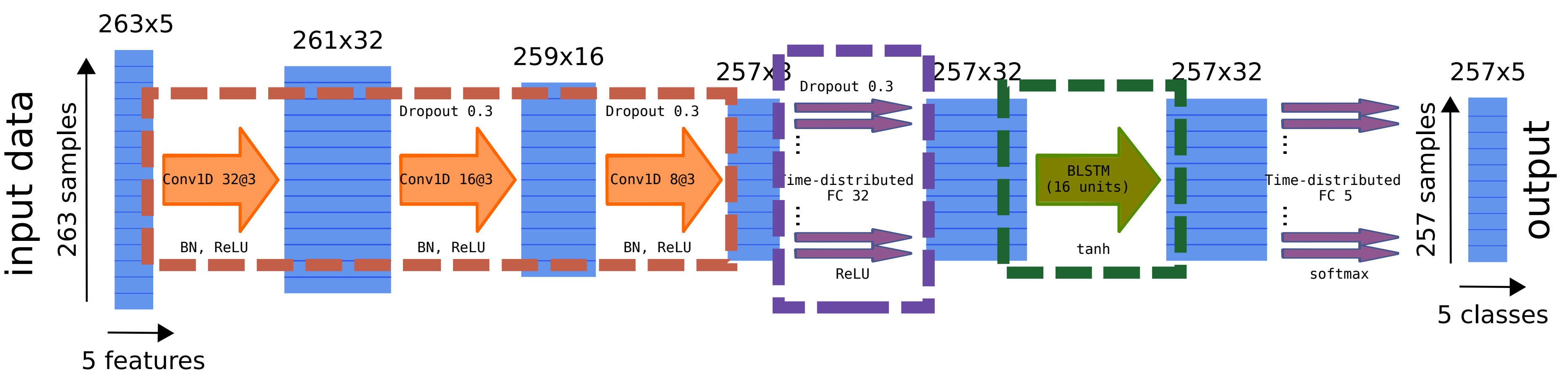
We pre-extracted speed and direction features on different scales from raw gaze location sequences.



Our model

A typical combination of layers for deep sequence-to-sequence processing:

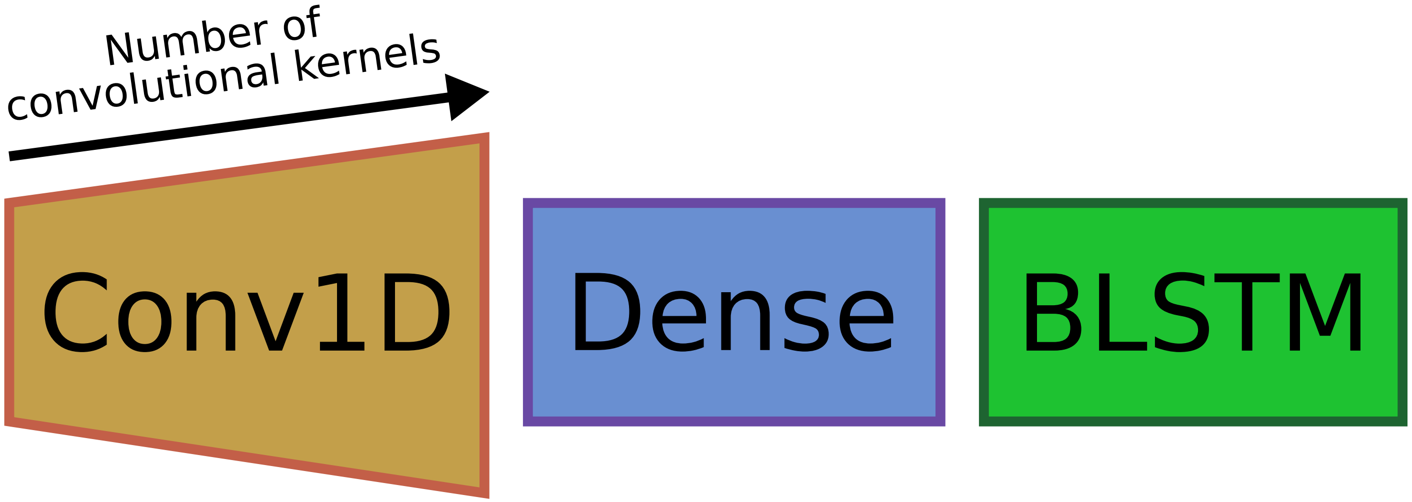
- Convolutional (1D in our case)
- Dense
- Long short-term memory – LSTM (bidirectional in our case)



Our “standard“ model

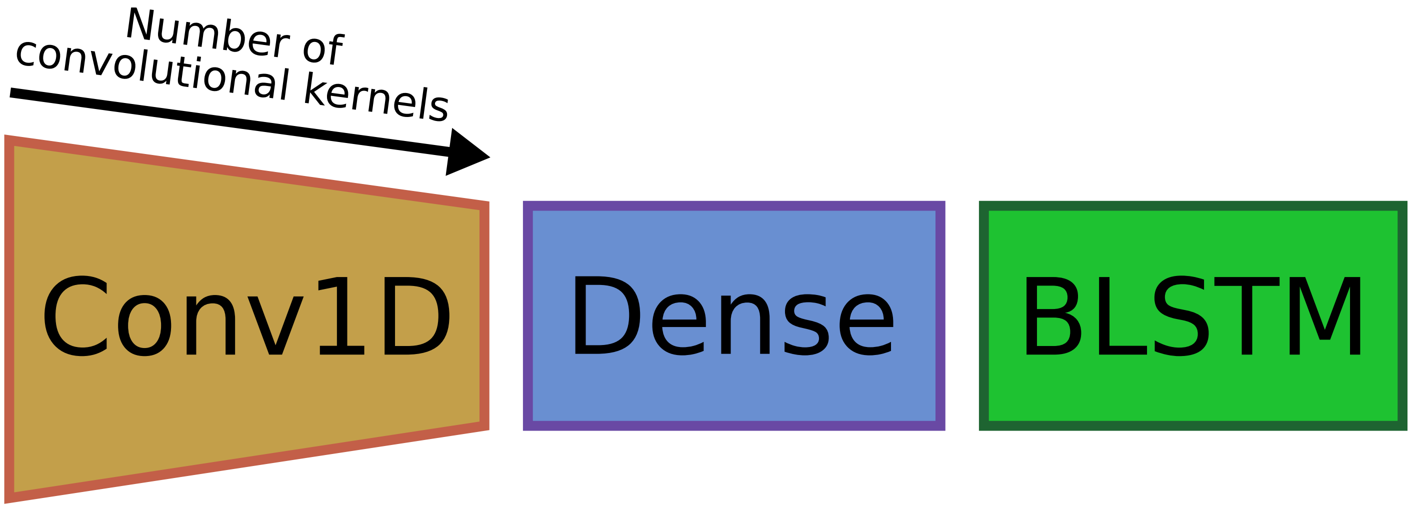
Our model: Convolutional block

- Fixed filter size
- Number of filters in each layer: increasing or decreasing?



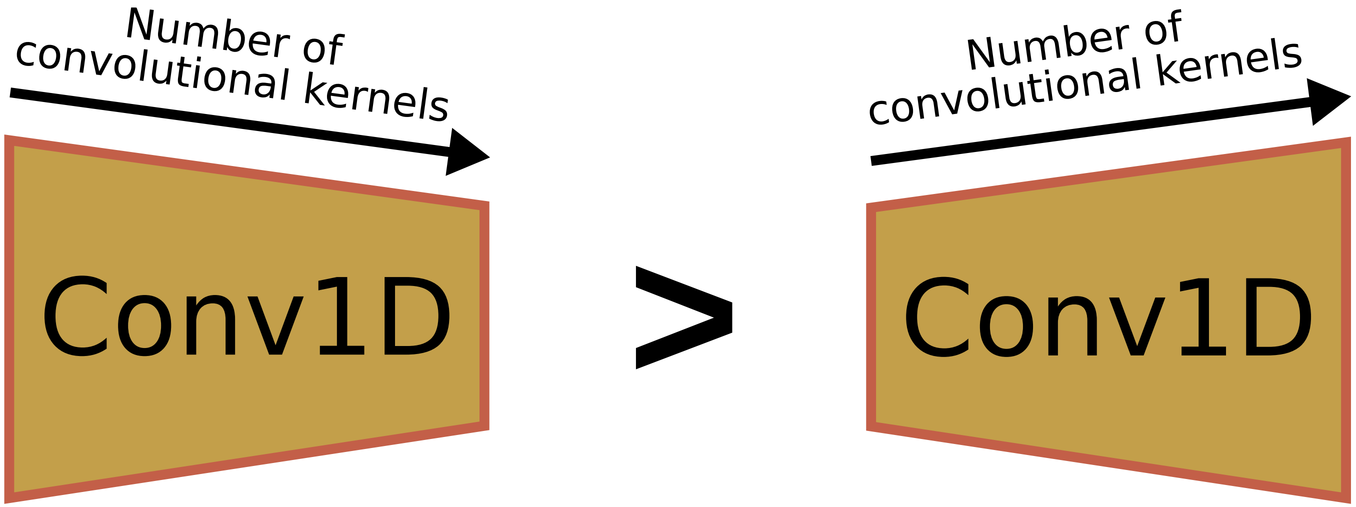
Our model: Convolutional block

- Fixed filter size
- Number of filters in each layer: increasing or decreasing?



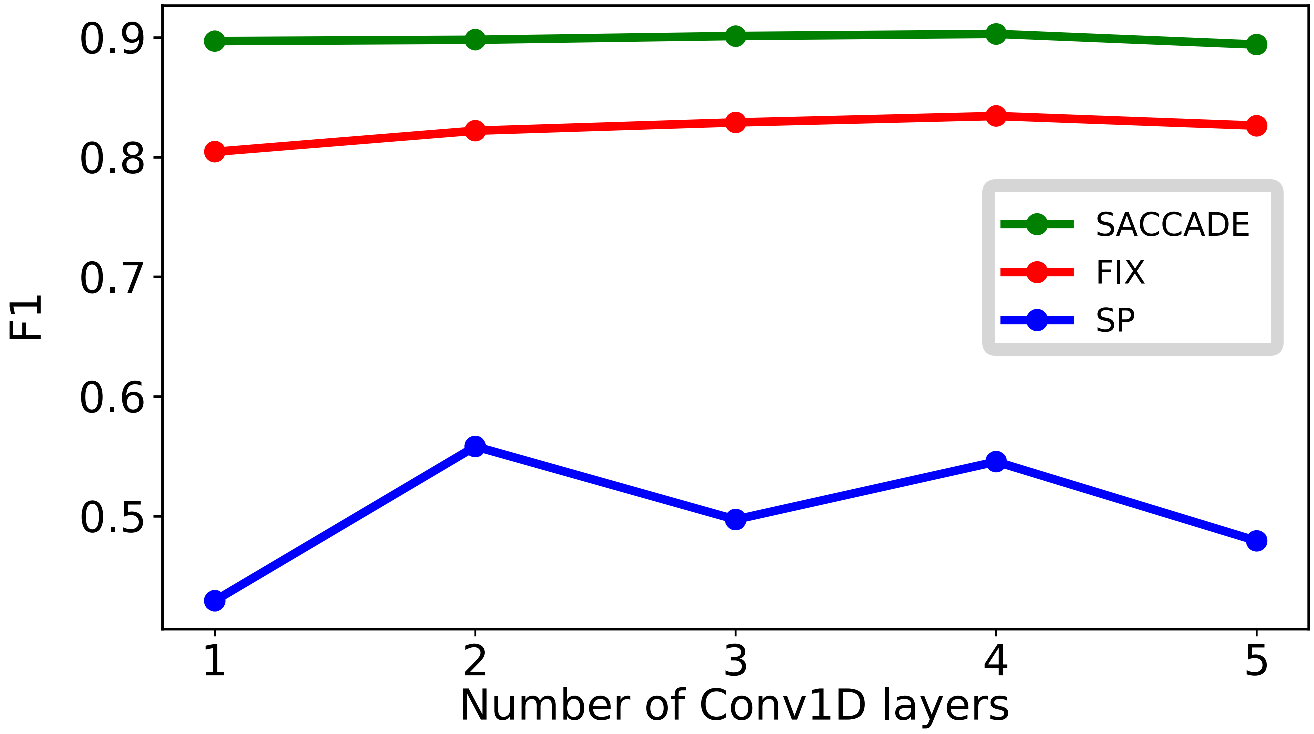
Our model: Convolutional block

- Fixed filter size
- Number of filters in each layer: increasing or **decreasing**?



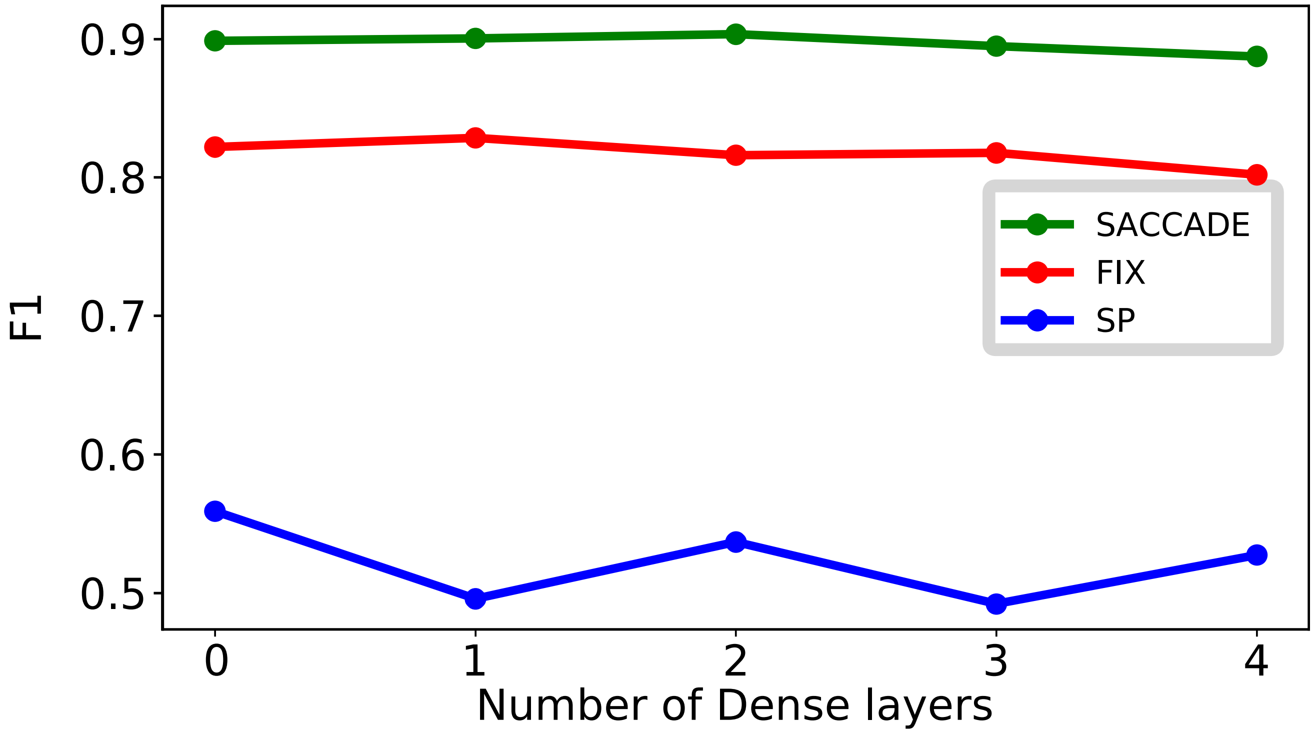
Our model: Convolutional block

- Fixed filter size
- Number of filters in each layer: increasing or **decreasing**?
- Varying number of layers



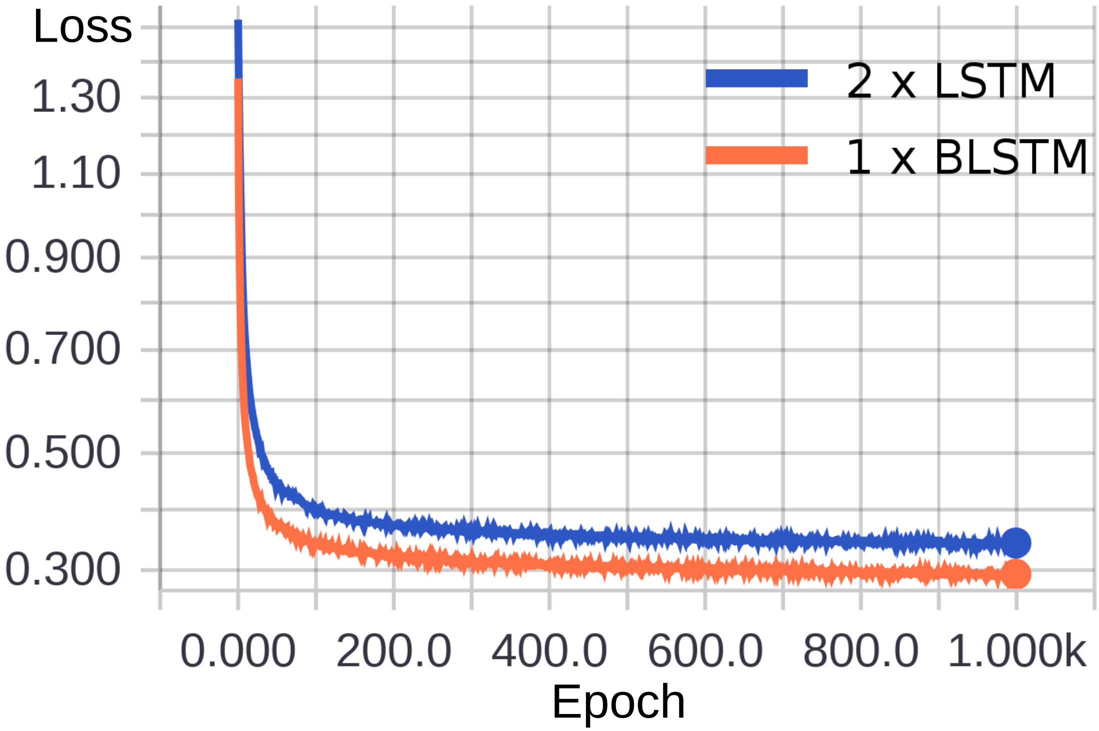
Our model: Dense block

- Fixed number of units
- Varying number of layers



Our model: Bidirectional LSTM block

- Do we really need bidirectional?

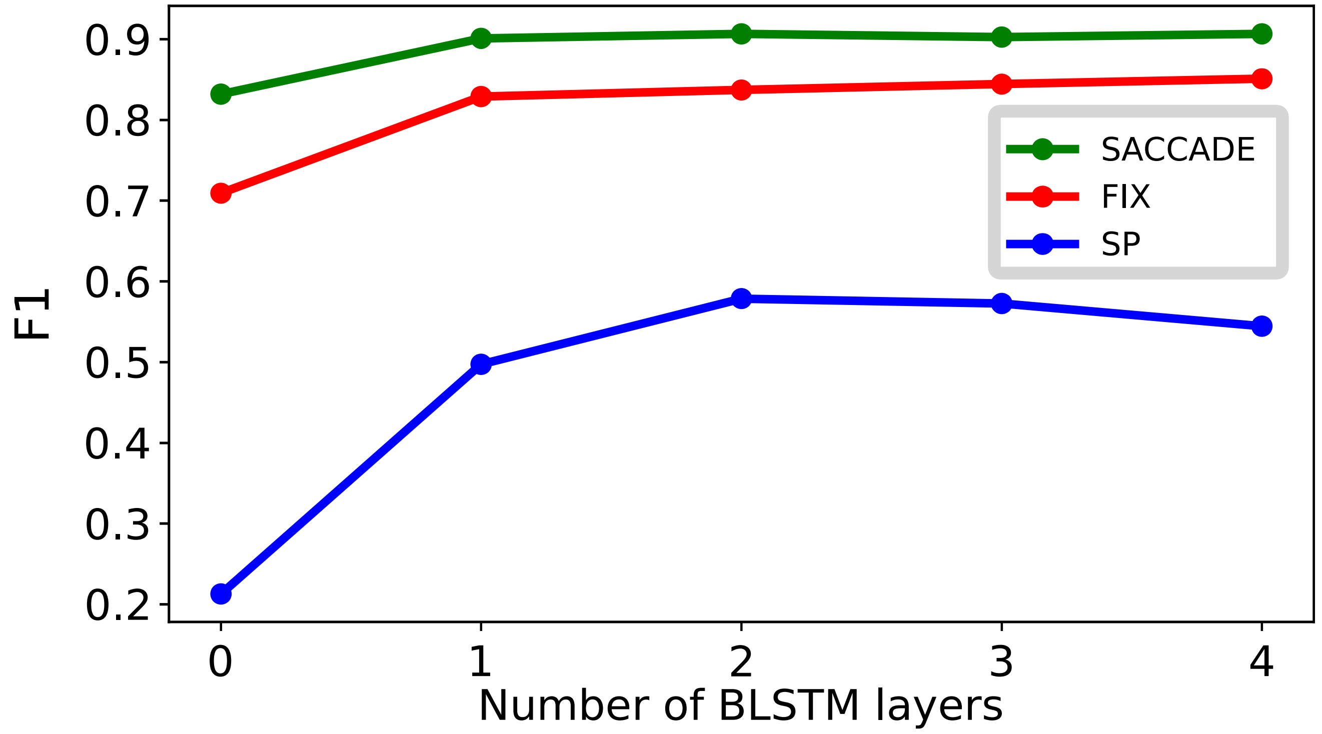


LSTM → BLSTM
performance increase
(in episode-level F1 scores)

Fixations	Saccades	Pursuits
10%	2%	17%

Our model: Bidirectional LSTM block

- Do we really need bidirectional?
- Varying number of layers
  (similar picture for various units counts in each layer)



Our model: Final architecture

- Four convolutional layers
- No dense layers
- Two stacked BLSTM layers

Ca. 13,000 trainable parameters, compared to ca. 10,000 in the “standard” model.

→ Increase model size with caution, depending in the data set size and diversity.

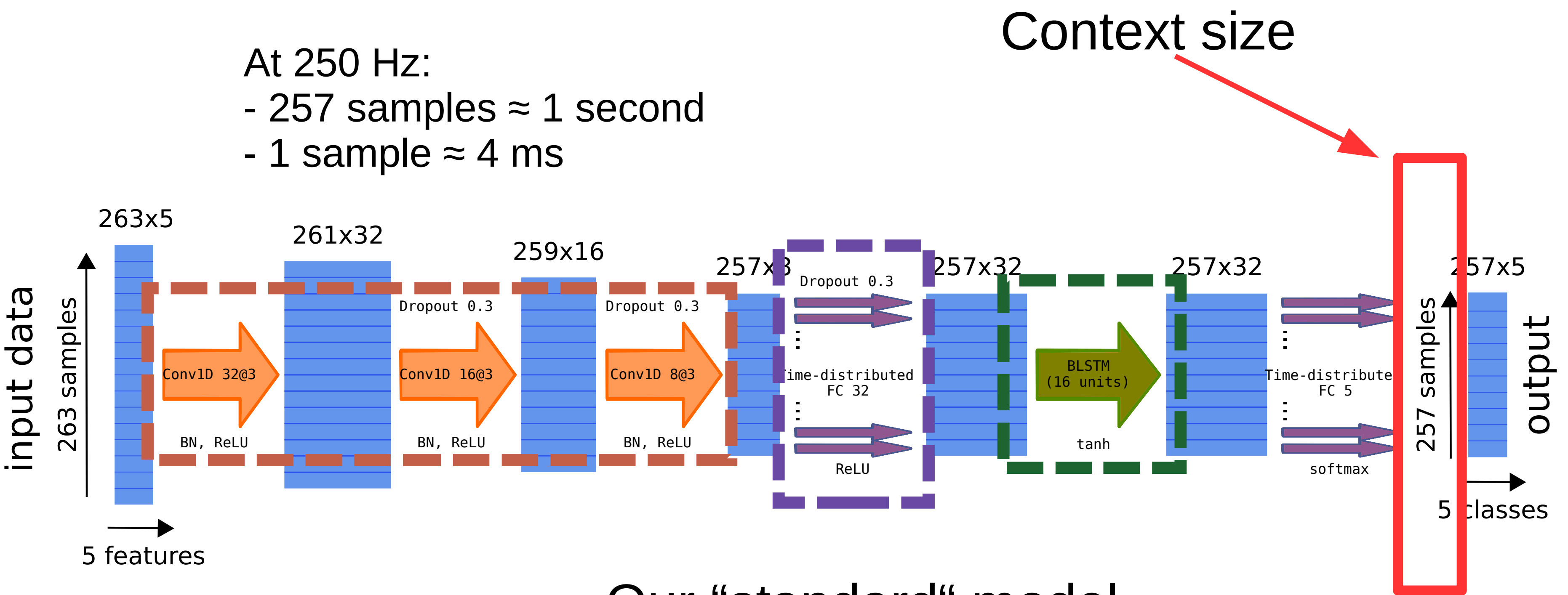
	Individual samples, F1 score			***Whole episodes, F1 score***		
	Fixations	Saccades	Pursuits	Fixations	Saccades	Pursuits
Final architecture	93.8%	89.6%	70.7%	91.5%	94.9%	62.9%
Increase (absolute)	*-0.2%*	*0.3%*	***0.4%***	*1.6%*	*0.2%*	***3.3%***

Our model: Final architecture

	Individual samples, F1 score			*Whole episodes, F1 score*		
	Fixations	Saccades	Pursuits	Fixations	Saccades	Pursuits
Final model	**93.8%**	**89.6%**	**70.7%**	**91.5%**	**94.9%**	**62.9%**
[Agtzidis et al. 2016]	88.6%	86.4%	64.6%	81.0%	88.4%	52.7%
I-VMP (optimized)	90.9%	68.0%	58.1%	79.2%	81.5%	53.1%
[Larsson et al. 2015]	91.2%	86.1%	45.9%	87.3%	88.4%	39.2%
[Berg et al. 2009]	88.3%	69.7%	42.2%	88.6%	85.6%	42.4%
Increase over state of the art (absolute)	**1.9%**	**3.2%**	**6.1%**	**1.3%**	**6.5%**	**9.8%**

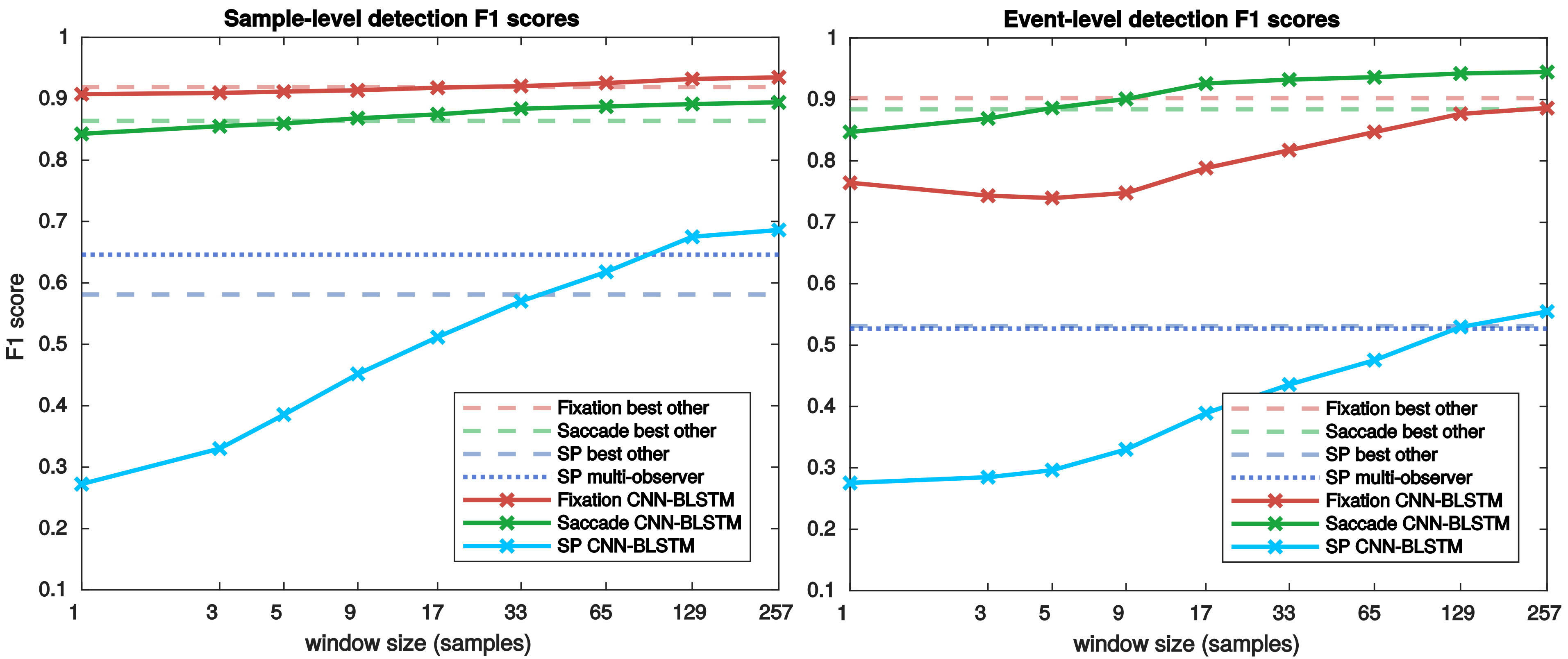
Our model: Influence of context size

At 250 Hz:
- 257 samples ≈ 1 second
- 1 sample ≈ 4 ms



Our “standard“ model

Our model: Influence of context size



(5 speed features used as input, varying context size, “standard” architecture)

Conclusions

- Compared to 12 literature models, our deep sequence-to-sequence solution performs best for each eye movement type
- Smooth pursuit is still trickier to detect than fixations and saccades
- Smooth pursuit benefits the most from
  - context size increase
  - architecture improvements

Thank you
for your attention!

Original data, benchmark, related publications:

http://michaeldorr.de/smoothpursuit

Code, data:

https://bit.ly/2MzbN9w

Thank you
for your attention!

Original data, benchmark, related publications:

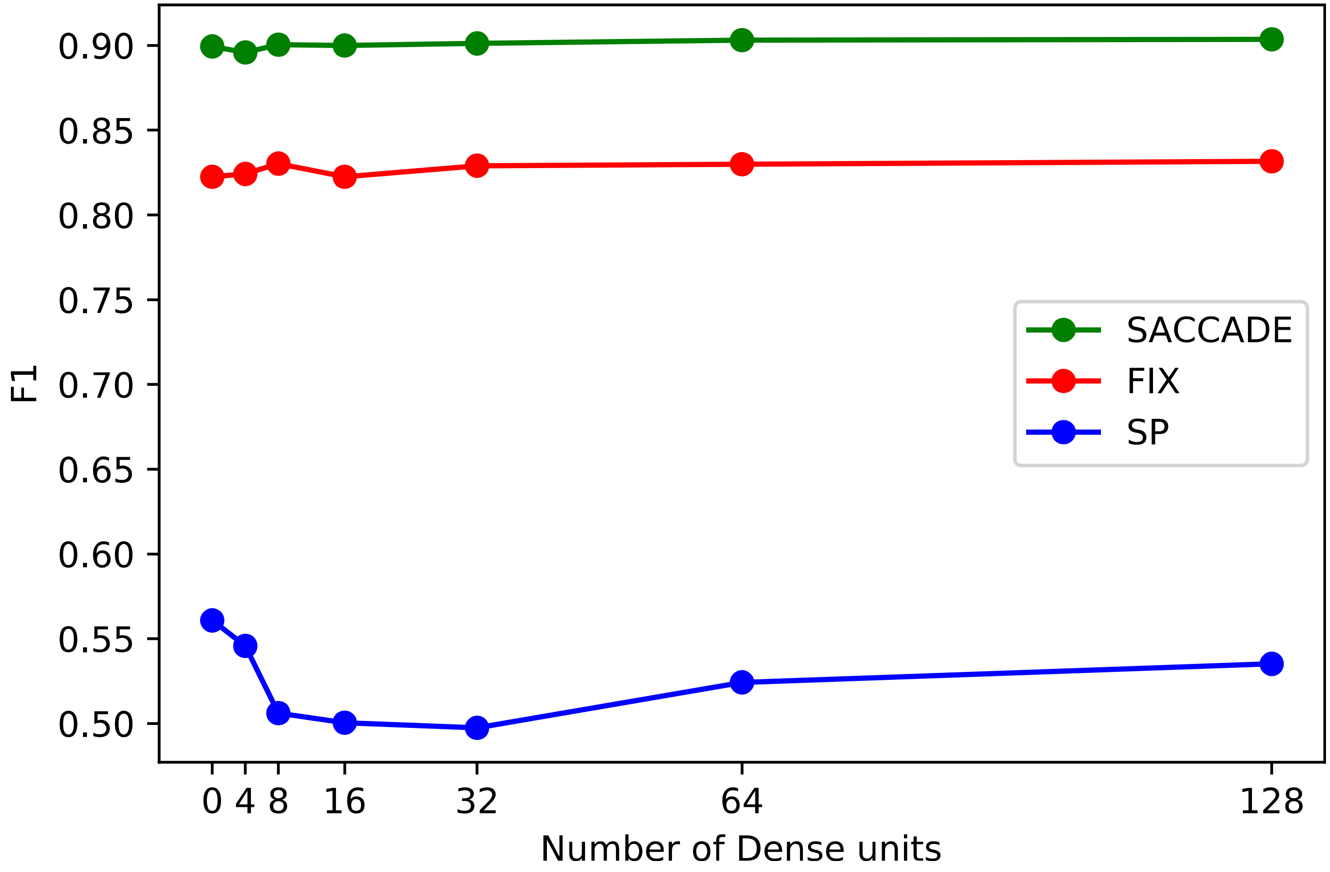
http://michaeldorr.de/smoothpursuit

Code, data:

https://bit.ly/2MzbN9w

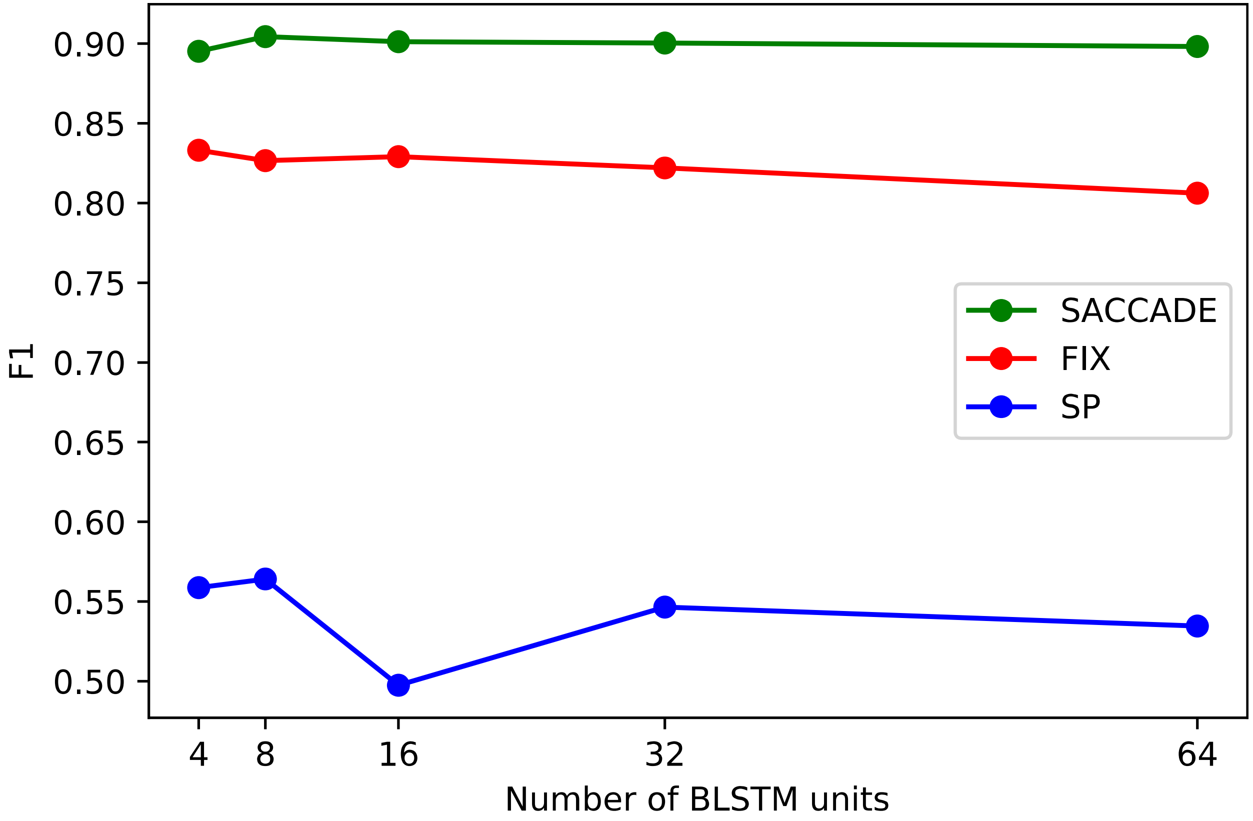
Our model: Dense block

- Varying number of units



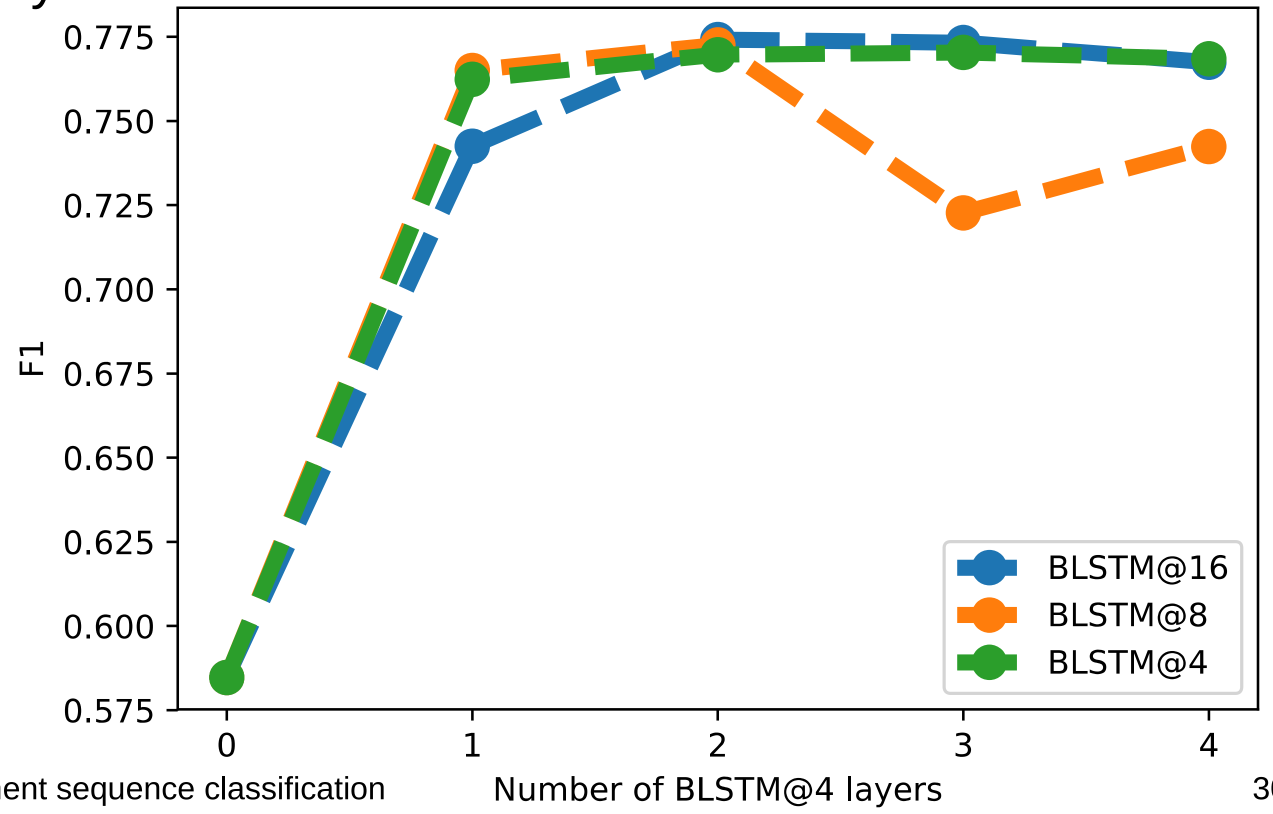
Our model: Bidirectional LSTM block

- Do we really need bidirectional?
- Varying number of units



Our model: Bidirectional LSTM block

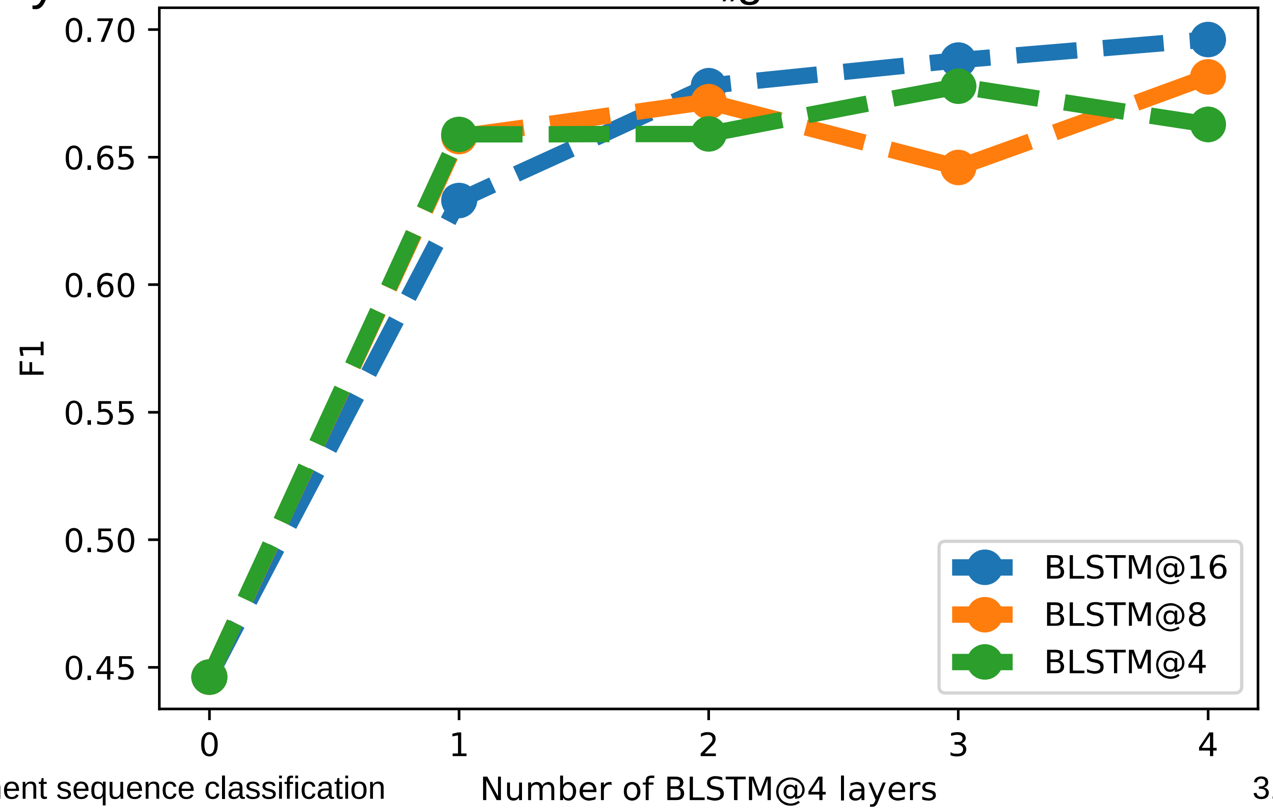
- Do we really need bidirectional?
- Varying number of units
- Varying number of layers



Our model: Bidirectional LSTM block

- Do we really need bidirectional?
- Varying number of units
- Varying number of layers

Event-level evaluation with
a threshold for „good" event matches



Our model: Final architecture

- Four convolutional layers
- No dense layers
- Two stacked BLSTM layers

Compared to the "standard" architecture
(at ca. 1s context windows):

	Individual samples, F1			***Whole episodes, IoU >= 0.5, F1***		
	Fixations	Saccades	Pursuits	Fixations	Saccades	Pursuits
Increase (absolute)	0.2%	0.3%	**0.4%**	1.5%	0.5%	**5.8%**
Final score	93.8%	89.6%	70.7%	88.2%	92.9%	54.2%