# A Light Transformer-Based Architecture for Handwritten Text Recognition

Killian Barrere Yann Soullard Aurélie Lemaitre Bertrand Coüasnon

Document Analysis Systems (DAS), La Rochelle
24th May 2022

IntuiDoc research team, Univ. Rennes, CNRS, IRISA, France

# State of the Art: From RNN to Transformer

## Usual approaches: Convolutional Recurrent Neural Networks (CRNN)

- Convolutional layers + recurrent layers

$\Rightarrow$ Lack of parallelism / **slow training speed**

# State of the Art: From RNN to Transformer

## Usual approaches: Convolutional Recurrent Neural Networks (CRNN)

- Convolutional layers + recurrent layers

$\Rightarrow$ Lack of parallelism / **slow training speed**

## Fully Convolutional Networks [Ingle et al. 2019, Yousef et al. 2020, Coquenet et al. 2021]

- Composed of convolutional layers, no recurrent layers

$\Rightarrow$ Faster training speed, but might be **hard to learn long-range contexts**

# State of the Art: From RNN to Transformer

## Usual approaches: Convolutional Recurrent Neural Networks (CRNN)

- Convolutional layers + recurrent layers

$\Rightarrow$ Lack of parallelism / **slow training speed**

## Fully Convolutional Networks [Ingle et al. 2019, Yousef et al. 2020, Coquenet et al. 2021]

- Composed of convolutional layers, no recurrent layers

$\Rightarrow$ Faster training speed, but might be **hard to learn long-range contexts**

## Multi-Head Attention (Transformer layers) [Vaswani et al. 2017]

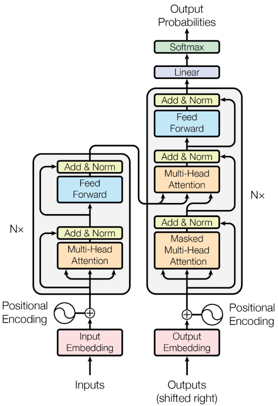
- Able to **learn long-range context**
- Strong **parallelism**

$\Rightarrow$ **Good alternative** but **require a lot of training data**

# Transformer for Handwritten Text Recognition

## Existing approaches [Kang et al. 2020, Singh et al. 2021]

- Transformer layers to model the language
- **Big architectures** to obtain state-of-the-art results




Original Transformer
[Vaswani et al. 2017]

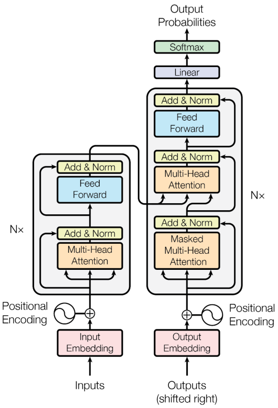
# Transformer for Handwritten Text Recognition

## Existing approaches [Kang et al. 2020, Singh et al. 2021]

- Transformer layers to model the language
- **Big architectures** to obtain state-of-the-art results

## Problem

- Require **a lot of data** to be trained
- Few annotated data in handwritten recognition (10k lines)
- ⇒ **Additional data** to perform well




Original Transformer [Vaswani et al. 2017]

# Transformer for Handwritten Text Recognition

## Existing approaches [Kang et al. 2020, Singh et al. 2021]

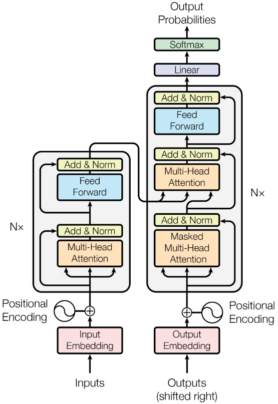
- Transformer layers to model the language
- **Big architectures** to obtain state-of-the-art results

## Problem

- Require **a lot of data** to be trained
- Few annotated data in handwritten recognition (10k lines)
- $\Rightarrow$ **Additional data** to perform well

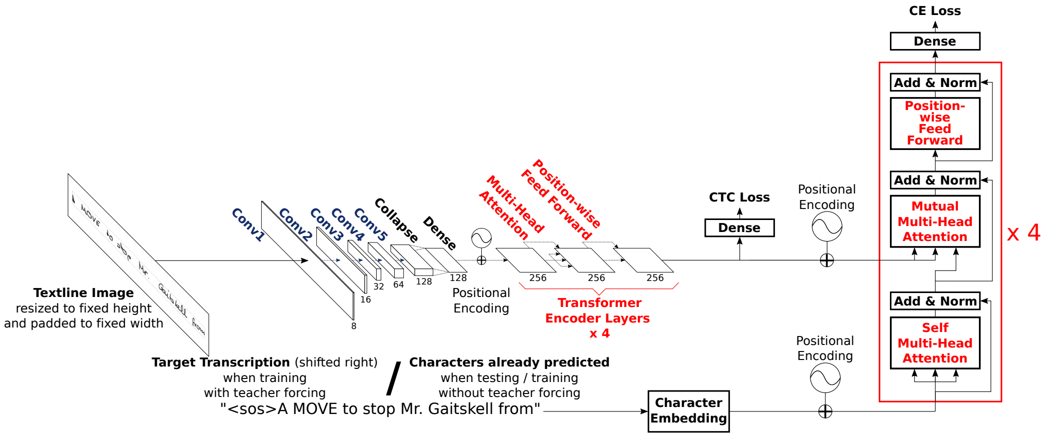
## Our proposition

- **Light architecture to perform well with few data**
- Hybrid loss to ease the training

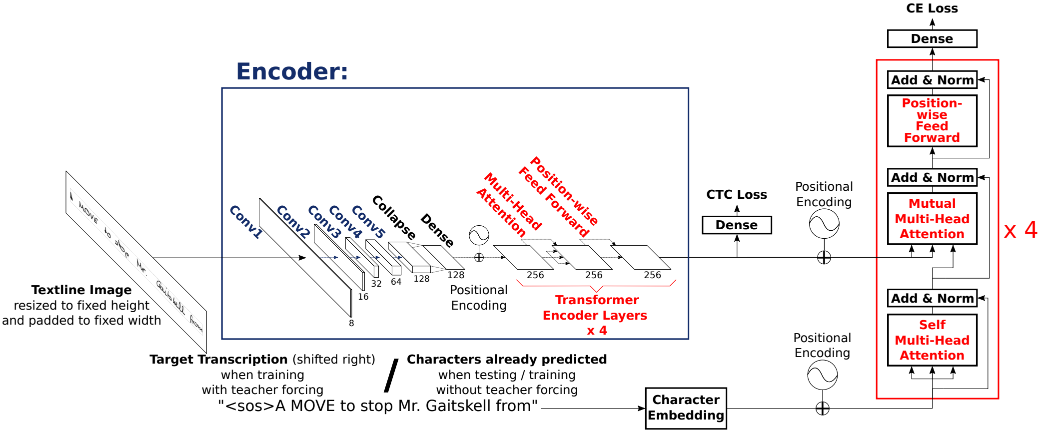



Original Transformer [Vaswani et al. 2017]

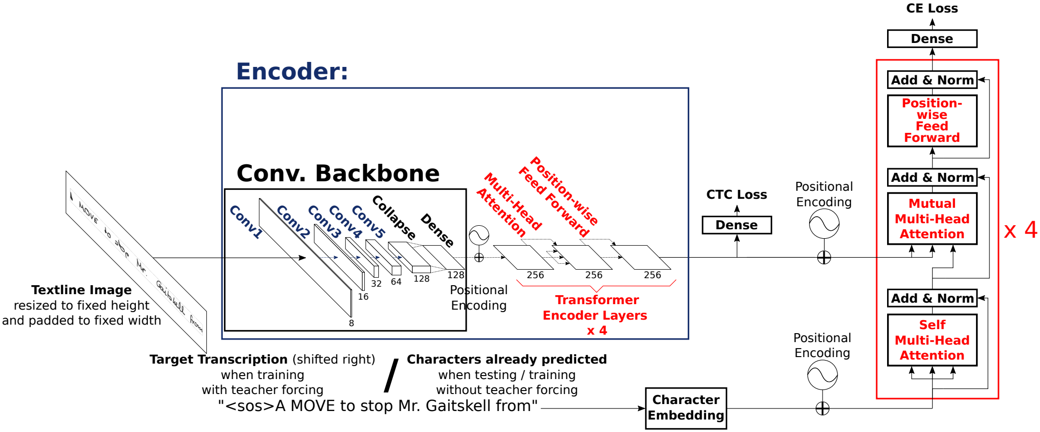
# Our Light Transformer Architecture



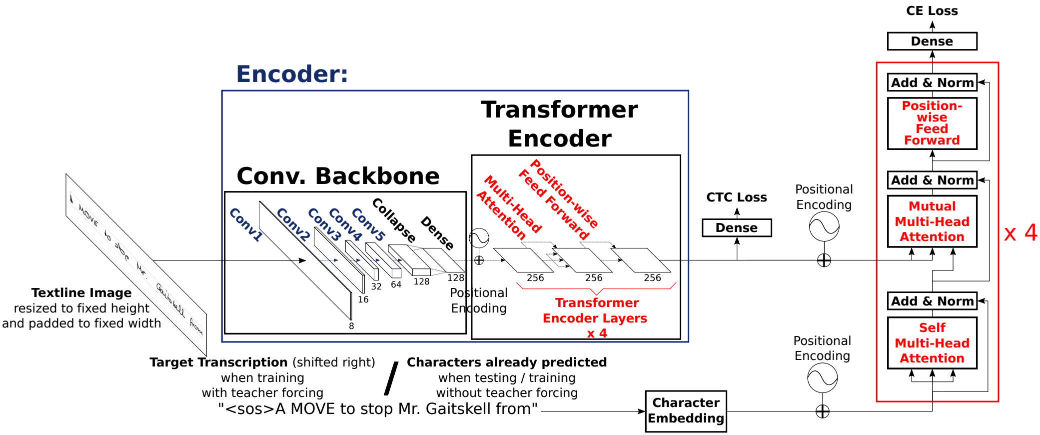
# Our Light Transformer Architecture



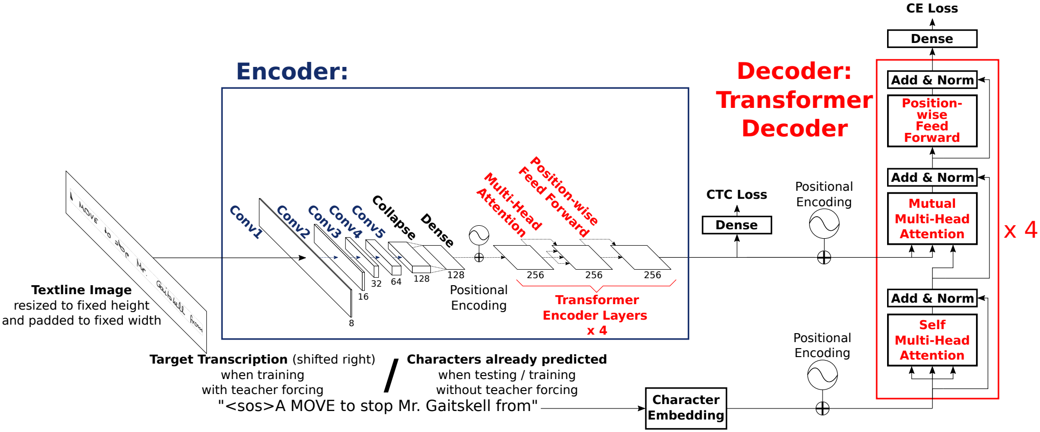
# Our Light Transformer Architecture



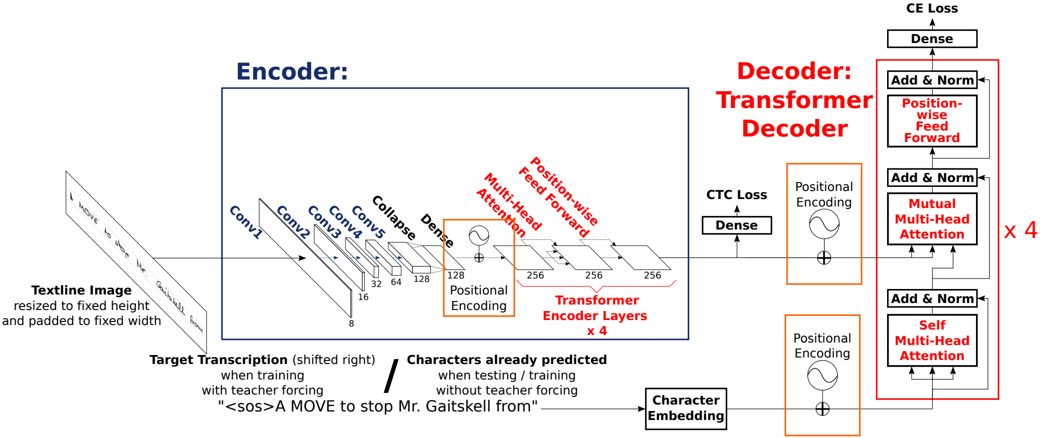
# Our Light Transformer Architecture



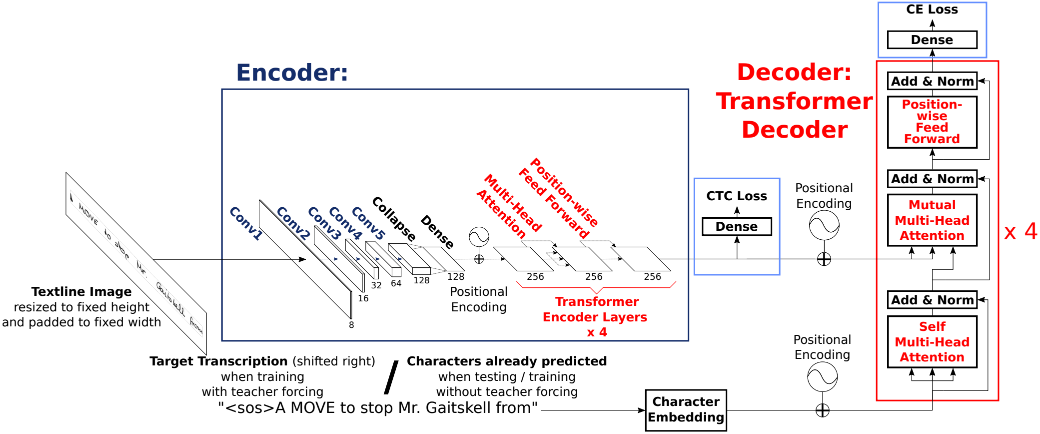
# Our Light Transformer Architecture



# Our Light Transformer Architecture



# Our Light Transformer Architecture

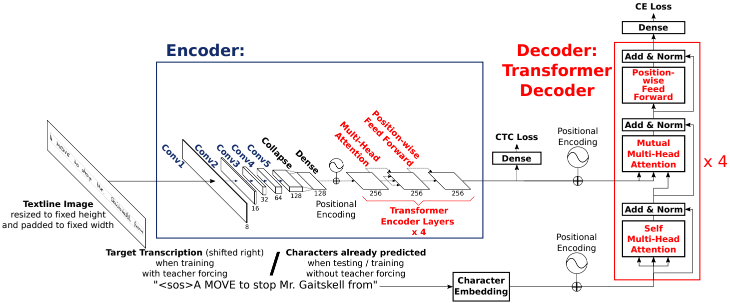


# A Light Architecture

## How to make a smaller Transformer

**Convolutional backbone**

Big backbone (i.e. ResNet18)



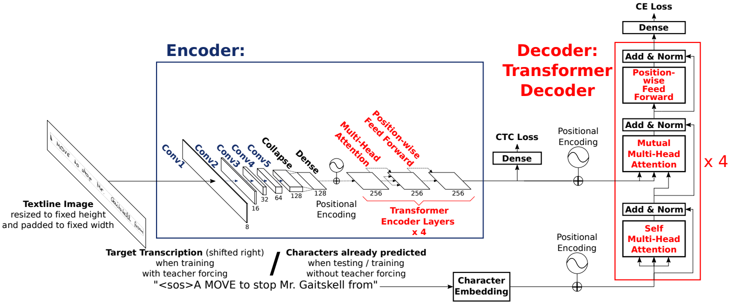
# A Light Architecture

## How to make a smaller Transformer

**Convolutional backbone**

~~Big backbone (i.e. ResNet18)~~

⇒ Only 5 convolutional layers



# A Light Architecture

## How to make a smaller Transformer

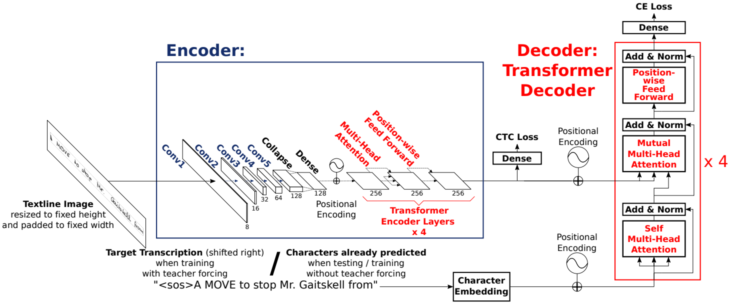
**Convolutional backbone**

~~Big backbone (i.e. ResNet18)~~

$\Rightarrow$ Only 5 convolutional layers

**Neurons in Transformer layers**

Up to 1,024 neurons



# A Light Architecture

## How to make a smaller Transformer

**Convolutional backbone**

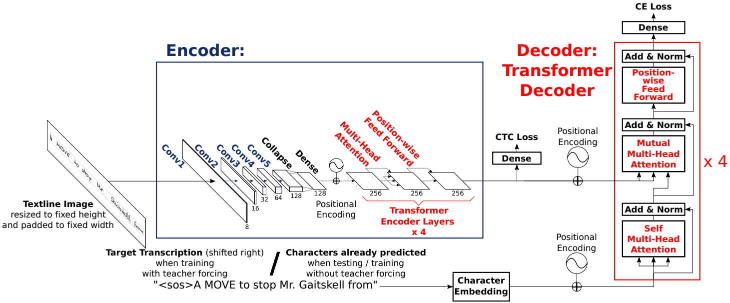
~~Big backbone (i.e. ResNet18)~~

⇒ Only 5 convolutional layers

**Neurons in Transformer layers**

~~Up to 1,024 neurons~~

⇒ Only 256 neurons



# A Light Architecture

## How to make a smaller Transformer

**Convolutional backbone**

~~Big backbone (i.e. ResNet18)~~

$\Rightarrow$ Only 5 convolutional layers

**Neurons in Transformer layers**

~~Up to 1,024 neurons~~

$\Rightarrow$ Only 256 neurons

## In total

100M parameters

# A Light Architecture

## How to make a smaller Transformer

**Convolutional backbone**

~~Big backbone (i.e. ResNet18)~~

$\Rightarrow$ Only 5 convolutional layers

**Neurons in Transformer layers**

~~Up to 1,024 neurons~~

$\Rightarrow$ Only 256 neurons

## In total

~~100M parameters~~

$\Rightarrow$ **6.9M parameters**

# A Light Architecture

## How to make a smaller Transformer

**Convolutional backbone**

~~Big backbone (i.e. ResNet18)~~

$\Rightarrow$ Only 5 convolutional layers

**Neurons in Transformer layers**

~~Up to 1,024 neurons~~

$\Rightarrow$ Only 256 neurons

## In total

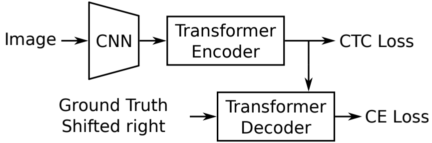
~~100M parameters~~

$\Rightarrow$ **6.9M parameters**

## Potential benefits

- **Faster to train** compared to other Transformer-based architecture
- **Does not require additional data** to be trained efficiently

# Hybrid Loss

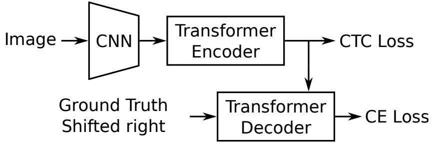



$$\mathcal{L} = \lambda \cdot \mathcal{L}_{CTC} + (1 - \lambda) \cdot \mathcal{L}_{CE}$$

## Hybrid loss [Michael et al. 2019]

- Connectionist Temporal Classification (CTC) for the Encoder
- Cross Entropy (CE) for the Decoder

# Hybrid Loss




$$\mathcal{L} = \lambda \cdot \mathcal{L}_{CTC} + (1 - \lambda) \cdot \mathcal{L}_{CE}$$

## Hybrid loss [Michael et al. 2019]

- Connectionist Temporal Classification (CTC) for the Encoder
- Cross Entropy (CE) for the Decoder

## Potential benefits

- Help to train deep layers with gradients from both losses
- Faster convergence

# Outline of the Experiments

## Experiments presented

1. Ablation Study
   - Transformer layers
   - Decoder
2. Architecture Size
3. Hybrid loss
4. Comparison with state-of-the-art methods

# Outline of the Experiments

## Experiments presented

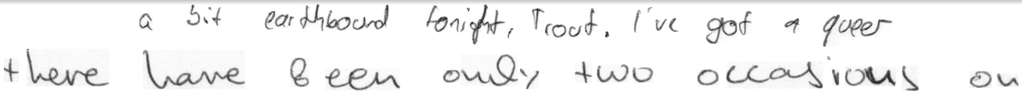
1. Ablation Study
   - Transformer layers
   - Decoder
2. Architecture Size
3. Hybrid loss
4. Comparison with state-of-the-art methods

- Results with and without synthetic data (to compare fairly with others)

# Data used

## Real data, without additional data

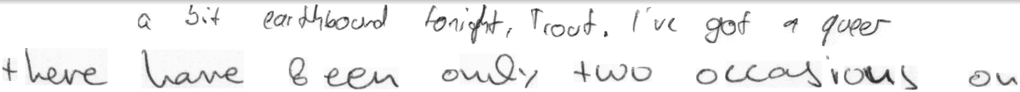
- IAM dataset (modern English, 10,363 lines, 76k words)
- Data augmentation techniques



# Data used

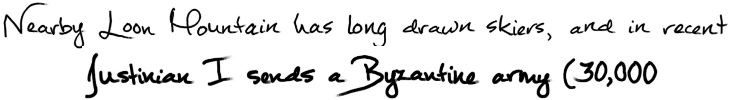
## Real data, without additional data

- IAM dataset (modern English, 10,363 lines, 76k words)
- Data augmentation techniques

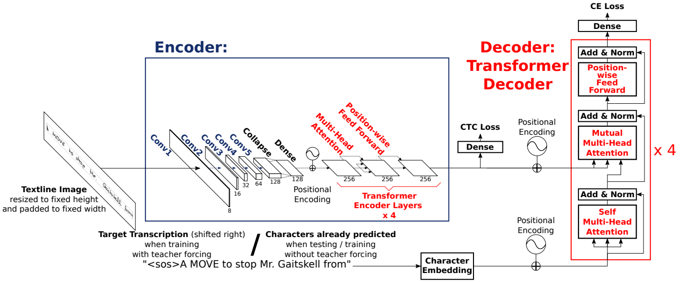



## Our synthetic data (to compare with other transformers)

- Articles from Wikipedia (21,350 articles, 66M words)
- Handwritten fonts (32 fonts)
- Random deformations / augmentations

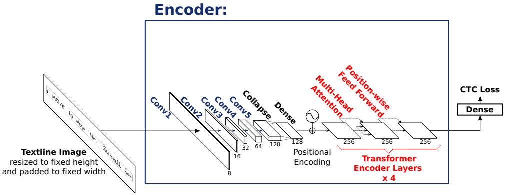


# Ablation Study: Transformer Layers instead of Recurrent Layers



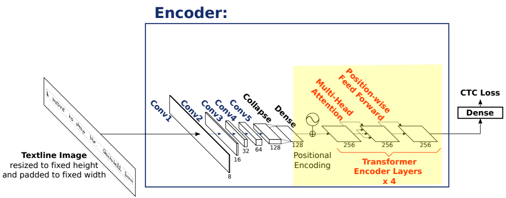

**Without the decoder**

# Ablation Study: Transformer Layers instead of Recurrent Layers




## Without the decoder

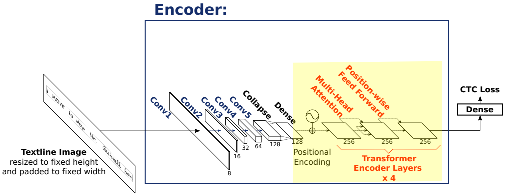
# Ablation Study: Transformer Layers instead of Recurrent Layers




## Without the decoder

- CRNN with Transformer instead of recurrent layers

# Ablation Study: Transformer Layers instead of Recurrent Layers




| Architecture | # params. | IAM | | IAM + Synth. Data | |
|---|---|---|---|---|---|
| | | CER (%) | WER (%) | CER (%) | WER (%) |
| CRNN (Baseline) | 1.7M | 6.14 | 23.26 | | |
| Our Encoder only | 3.2M | 5.93 | 22.82 | | |

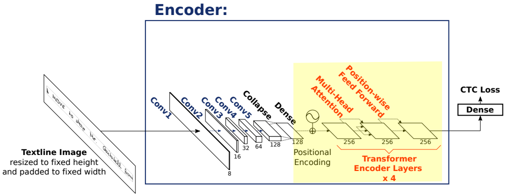
**Without the decoder**

- CRNN with Transformer instead of recurrent layers

**Recurrent $\Rightarrow$ Transformer**

- More parameters
- Lower error rates
  - Better context

# Ablation Study: Transformer Layers instead of Recurrent Layers




| Architecture | # params. | IAM | | IAM + Synth. Data | |
|---|---|---|---|---|---|
| | | CER (%) | WER (%) | CER (%) | WER (%) |
| CRNN (Baseline) | 1.7M | 6.14 | 23.26 | 5.66 | 21.62 |
| Our Encoder only | 3.2M | 5.93 | 22.82 | 6.15 | 24.02 |

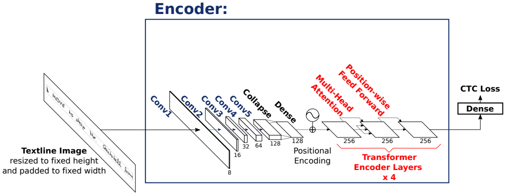
**Without the decoder**

- CRNN with Transformer instead of recurrent layers

**Recurrent ⇒ Transformer**

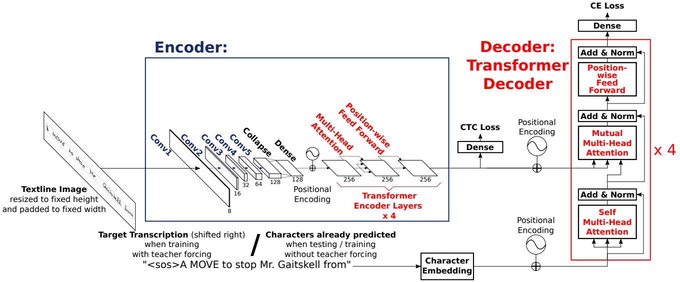
- More parameters
- Lower error rates
  - Better context
- Worse with synthetic data (may not generalize well)

# Ablation Study: Decoder



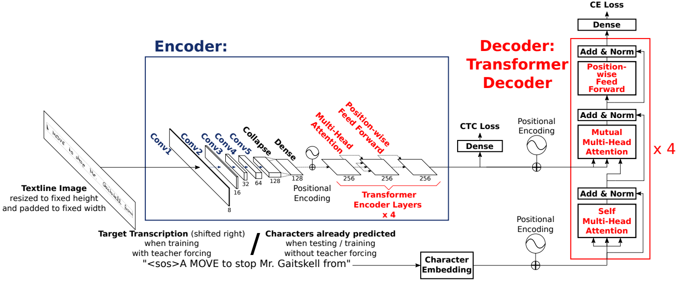

## With the decoder

# Ablation Study: Decoder




## With the decoder

# Ablation Study: Decoder



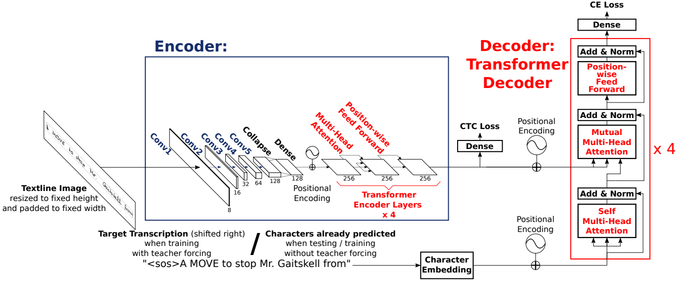

| Architecture | # params. | IAM CER (%) | IAM WER (%) | IAM + Synth. Data CER (%) | IAM + Synth. Data WER (%) |
|---|---|---|---|---|---|
| CRNN (Baseline) | 1.7M | 6.14 | 23.26 | | |
| Our Encoder only | 3.2M | 5.93 | 22.82 | | |
| Our Light Transformer | 6.9M | **5.70** | **18.86** | | |

## With the decoder

**Ability to model the language**

- Lower error rates
- Stronger impact on the WER

# Ablation Study: Decoder




| Architecture | # params. | IAM | | IAM + Synth. Data | |
|---|---|---|---|---|---|
| | | CER (%) | WER (%) | CER (%) | WER (%) |
| CRNN (Baseline) | 1.7M | 6.14 | 23.26 | 5.66 | 21.62 |
| Our Encoder only | 3.2M | 5.93 | 22.82 | 6.15 | 24.02 |
| Our Light Transformer | 6.9M | **5.70** | **18.86** | **4.76** | **16.31** |

## With the decoder

**Ability to model the language**

- Lower error rates
- Stronger impact on the WER

**Benefits more from synthetic data**

- More data to learn the language

# Benefits of Using a Light Architecture

**Different sizes** of our architecture

- **Light Transformer: 6.9M params.**
- Large Transformer: 28M params.

## Benefits of Using a Light Architecture

**Different sizes** of our architecture

- **Light Transformer: 6.9M params.**
- Large Transformer: 28M params.

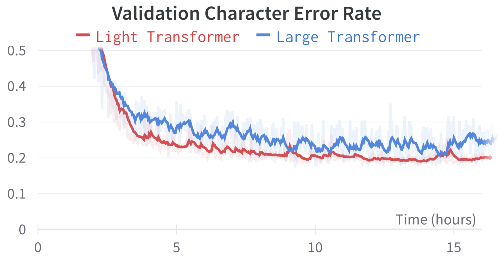
| Architecture | IAM | | IAM + Synth. Data | |
|---|---|---|---|---|
| | CER (%) | WER (%) | CER (%) | WER (%) |
| Our Light Transformer | **5.70** | **18.86** | **4.76** | **16.31** |
| Our Large Transformer | 5.79 | 19.67 | 4.87 | 17.67 |

- Our light architecture is **competitive**

# Benefits of Using a Light Architecture

**Different sizes** of our architecture

- **Light Transformer: 6.9M params.**
- Large Transformer: 28M params.




| Architecture | IAM | | IAM + Synth. Data | |
|---|---|---|---|---|
| | CER (%) | WER (%) | CER (%) | WER (%) |
| Our Light Transformer | **5.70** | **18.86** | **4.76** | **16.31** |
| Our Large Transformer | 5.79 | 19.67 | 4.87 | 17.67 |

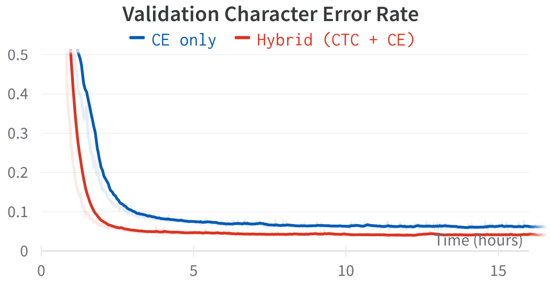
- Our light architecture is **competitive**
- Our light architecture might be **trained faster**

# Interest of the Hybrid Loss

- **CE only**: Cross-Entropy loss after the decoder
- **Hybrid**: CTC after the encoder and CE after the decoder

# Interest of the Hybrid Loss

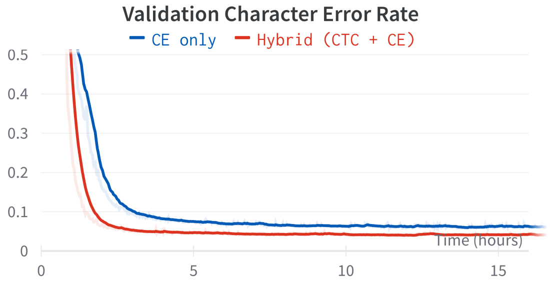
- **CE only**: Cross-Entropy loss after the decoder
- **Hybrid**: CTC after the encoder and CE after the decoder




- Faster convergence

# Interest of the Hybrid Loss

- **CE only**: Cross-Entropy loss after the decoder
- **Hybrid**: CTC after the encoder and CE after the decoder

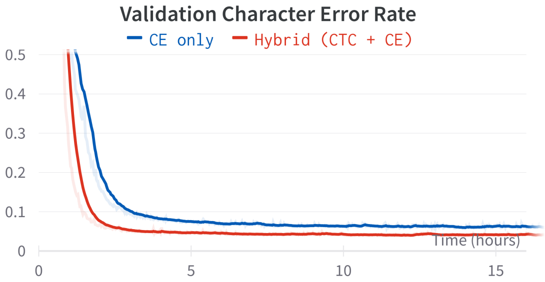



| Loss Function(s) | IAM | | IAM + Synth. Data | |
|---|---|---|---|---|
| | CER (%) | WER (%) | CER (%) | WER (%) |
| CE only | 10.29 | 26.36 | | |
| Hybrid (CTC + CE) | **5.70** | **18.86** | | |

- Faster convergence
- Crucial with few data

# Interest of the Hybrid Loss

- **CE only**: Cross-Entropy loss after the decoder
- **Hybrid**: CTC after the encoder and CE after the decoder




| Loss Function(s) | IAM | | IAM + Synth. Data | |
|---|---|---|---|---|
| | CER (%) | WER (%) | CER (%) | WER (%) |
| CE only | 10.29 | 26.36 | 6.76 | 19.62 |
| Hybrid (CTC + CE) | **5.70** | **18.86** | **4.76** | **16.31** |

- Faster convergence
- Crucial with few data
- Important with synthetic data

# Comparison with the state of the art

| Model Encoder | # params. | IAM CER (%) | IAM + Synth. Data CER (%) |
|---|---|---|---|
| CRNN + LSTM [Michael et al. 2019] | | 5.24 | |
| FCN [Yousef et al. 2020] | 3.4M | **4.9** | |
| VAN (line level) [Coquenet et al. 2022] | 1.7M | 4.95 | |
| Transformer [Kang et al. 2020] | 100M | 7.62 | **4.67** |
| FPHR Transformer [Singh et al. 2021] | 28M | | 6.5 |
| Forward Transformer [Wick et al. 2021] | 13M | 6.03 | |
| Bidi. Transformer [Wick et al. 2021] | 27M | 5.67 | |
| Our Light Transformer-based | 6.9M | 5.70 | 4.76 |

# Comparison with the state of the art

Compared with other Transformers:

| Model Encoder | # params. | IAM CER (%) | IAM + Synth. Data CER (%) |
|---|---|---|---|
| CRNN + LSTM [Michael et al. 2019] | | 5.24 | |
| | 3.4M | **4.9** | |
| ...2022] | 1.7M | 4.95 | |
| Transformer [Kang et al. 2020] | 100M | 7.62 | **4.67** |
| FPHR Transformer [Singh et al. 2021] | 28M | | 6.5 |
| Forward Transformer [Wick et al. 2021] | 13M | 6.03 | |
| Bidi. Transformer [Wick et al. 2021] | 27M | 5.67 | |
| Our Light Transformer-based | 6.9M | 5.70 | 4.76 |

## Comparison with the state of the art

| Model Encoder | # params. | IAM CER (%) | IAM + Synth. Data CER (%) |
|---|---|---|---|
| CRNN + LSTM [Michael et al. 2019] | | 5.24 | |
| FCN [Yousef et al. 2020] | 3 | | |
| VAN (line level) [Coquenet et al. 2022] | 1.7 | | |
| Transformer [Kang et al. 2020] | 100M | 7.62 | **4.67** |
| FPHR Transformer [Singh et al. 2021] | 28M | | 6.5 |
| Forward Transformer [Wick et al. 2021] | 13M | 6.03 | |
| Bidi. Transformer [Wick et al. 2021] | 27M | 5.67 | |
| Our Light Transformer-based | 6.9M | 5.70 | 4.76 |

**Low error rates without additional data**

# Comparison with the state of the art

| Model Encoder | # params. | IAM CER (%) | IAM + Synth. Data CER (%) |
|---|---|---|---|
| CRNN + LSTM [Michael et al. 2019] | | 5.24 | |
| FCN [Yousef et al. 2020] | 3.4M | **4.9** | |
| VAN (line level) [Coquenet et al. 2022] | 1.7M | 4.95 | |
| Transformer [Kang et al. 2020] | 100M | 7.62 | **4.67** |
| FPHR Transformer [Singh et al. 2021] | 28M | | 6.5 |
| Forward Transformer [Wick et al. 2021] | 13M | 6.03 | |
| Bidi. Transformer [Wick et al. 2021] | 27M | 5.67 | |
| Our Light Transformer-based | 6.9M | 5.70 | 4.76 |

**State-of-the-art results with synthetic data**

# Comparison with the state of the art

| Model Encoder | # params. | IAM CER (%) | IAM + Synth. Data CER (%) |
|---|---|---|---|
| CRNN + LSTM [Michael et al. 2019] | | 5.24 | |
| FCN [Yousef et al. 2020] | | **4.9** | |
| VAN (line level) [Coquenet et al. 20 ] | | 4.95 | |
| Transformer [Kang et al. 2020] | 100M | 7.62 | **4.67** |
| FPHR Transformer [Singh et al. 2021] | 28M | | 6.5 |
| Forward Transformer [Wick et al. 2021] | 13M | 6.03 | |
| Bidi. Transformer [Wick et al. 2021] | 27M | 5.67 | |
| Our Light Transformer-based | 6.9M | 5.70 | 4.76 |

While being
**a light Transformer**

# Comparison with the state of the art

| Model Encoder | # params. | IAM CER (%) | IAM + Synth. Data CER (%) |
|---|---|---|---|
| CRNN + LSTM [Michael et al. 2019] | | 5.24 | |
| FCN [Yousef et al. 2020] | 3.4M | **4.9** | |
| VAN (line level) [Coquenet et al. 2022] | 1.7M | 4.95 | |
| Transformer [Kang et al. 2020] | 100M | 7.62 | **4.67** |
| FPHR Transformer [Singh et al. 2021] | 28M | | 6.5 |
| Forward Transformer [Wick et al. 2021] | 13M | 6.03 | |
| Bidi. Transformer [Wick et al. 2021] | 27M | 5.67 | |
| Our Light Transformer-based | 6.9M | 5.70 | 4.76 |

# Conclusion

## Our Contribution

A **light Transformer architecture**, trained with a **hybrid loss**

- **Faster to train** than other Transformers
- **Good results without additional data**
- **State-of-the-art results** with synthetic data

# Conclusion

## Our Contribution

A **light Transformer architecture**, trained with a **hybrid loss**

- **Faster to train** than other Transformers
- **Good results without additional data**
- **State-of-the-art results** with synthetic data

## Future Works: Historical Documents

- Ability of Transformers to **model the language** is crucial
- **Very few annotated data** $\Rightarrow$ our light Transformer architecture