2024

UM6P College of Computing

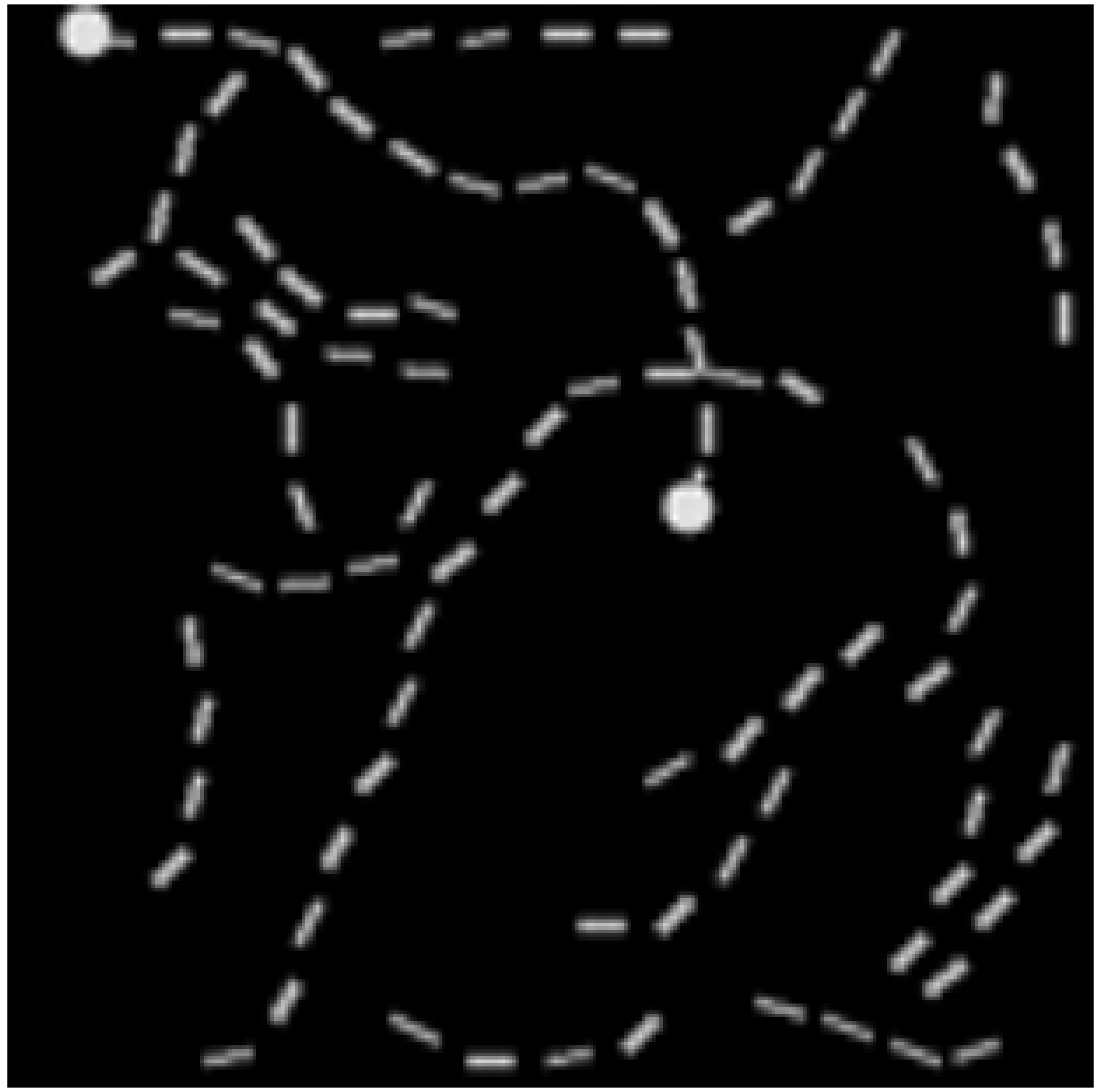
Long-range-arena Benchmarking on Pathfinder dataset

CHOUKRI Zakaria
ARNAOUI Basma
AIT MAGOURT Adnane

Supervised By:
Hamza ALAMI
Issam AIT YAHIA

Pathfinder Dataset

- Images used: 32x32 pixels
- Binary classification task: Identify if two points are connected by a dashed path.
- We used images from curv_contour_length_14: Includes images with long contours of length 14, presenting the most challenging task.
- Total of 200.000 images



Models evaluated

CHOUKRI Zakaria

Model	# Parameters	Accuracy	Training time	Efficiency
CNN Model	2226434	0.7968	3193.16 s	0.0068
DEIT Model	21666434	0.7883	10064.90 s	0.0051
Transformer model	2631490	0.5011	10138.11 s	0.0037

CNN Model

- Data Processing:
    - Dataset Loading
    - Dataset Splitting 0.8:0.2

- Model Architecture:
    - MobileNetV2 consists of a series of inverted residual blocks, with each block typically having three components:
        - Expansion layer (1x1 convolution): Expands the input channels.
        - Depthwise separable convolution: Applies convolution over individual channels (rather than all channels simultaneously).
        - Linear bottleneck: Reduces the channel size at the output of the block.

CNN Model

- Training:
  - The model is pretrained but we fine tune it
  - Optimizer: AdamW
  - Loss function: Binary cross entropy
  - Metrics: Training time, number of parameters, accuracy, efficiency score
- Conclusion:
  - this model performed really good which was expected, since CNNs are meant to perform great with images, but this is not the point of the project. Instead, this model was used just as a baseline to better understand the quality of the 2 other models.

DeiT Model

- Data Processing:
  - Dataset Loading
  - Dataset Splitting 0.8:0.2
  - Data Transformation: resize images to become 224x224

- Model Architecture:
  - The input image of size 224x224 is divided into non-overlapping patches of size 16x16
  - These patches are flattened into vectors and projected through a linear embedding layer
  - Transformer layers
  - Final Classification Head

DeiT Model

- Training:
  - The model is pretrained but we fine tune it
  - Optimizer: AdamW
  - Loss function: Binary cross entropy
  - Metrics: Training time, number of parameters, accuracy, efficiency score

- Conclusion:
  - this model performed good too, not better than the CNN model but close to it. However, it had one flaw, it didn’t treat the images like a sequence of 1024 pixels, but made patches of 16x16 pixels, so the sequences were only 256 of length. Though the results were overall good and practical, It still doesn’t stress test the transformer enough because of this fact. So the next model is a pure transformer that treats images a sequence of 1024 pixels.

Transformer Model

- Data Processing:
  - Dataset Loading
  - Dataset Splitting 0.8:0.2
  - Data Transformation: images are flattened into a sequence of 1024 pixels

- Model Architecture:
  - Input Embedding Layer and positional encoding
  - Transformer Encoder: Consists of multiple layers of multi-head self-attention and feedforward neural networks.
  - Classification Head

Transformer Model

- Training:
  - Optimizer: Adam
  - Loss function: Binary cross entropy
  - Metrics: Training time, number of parameters, accuracy, efficiency score

- Conclusion:
  - this model did not perform as well as the other two. But it is expected, since the long sequence length of 1024 is something that stress tests the model’s attention mechanism. It also avoids any bias introduced by preprocessing which can help in the training but it is not something we are interested in in our use case.

Thank you very much !