WEB3CLUBS FOUNDATION LIMITED

Course Instructor: DR. Cyprian Omukhwaya Sakwa
PHONE: +254723584205 Email: cypriansakwa@gmail.com

Foundational Mathematics for Web3 Builders

Implemented in RUST

Lecture 41

August 7, 2024

(2) Create a privacy-preserving image style transfer application using Concrete ML €10,000 $\approx$ Ksh $14,000,000$

- Concrete ML facilitates data scientists' usage of Fully Homomorphic Encryption (FHE) by automating the conversion of machine learning models into homomorphic variants.
- That is, Concrete ML is a tool or framework that simplifies the implementation of Fully Homomorphic Encryption for data scientists.
- FHE enables computations to be done on encrypted data without the need to decrypt it, ensuring data privacy.
- Style Transfer, a computer vision and graphics technology, generates a new image by combining the content of one image with the style of another image or group of images.
- The goal of style transfer is to create an image that keeps the original content while taking on the visual style of the other image(s).

18.3 Several neural network architectures that can be used for style transfer:

- Convolutional Neural Networks (CNNs), like VGG-16 and VGG-19.
- Encoder-Decoder Networks.
- Perceptual Loss Networks, which use a loss function based on feature maps from a pre-trained network.
- Transformers, especially Vision Transformers (ViTs).
- Neural Style Transfer (NST), which optimizes an image to match content and style features from a pre-trained CNN.

✓ Style transfer frequently entails processing personal images, such as photos of humans, which may contain sensitive data.

✓ Protecting the privacy of these photographs is critical to avoid illegal access or use.

- ✓ Unauthorized access to photos can result in a variety of security problems, including identity theft, phishing attempts, and other harmful behavior.
- Your task is to build a style transfer application using Concrete ML while protecting data privacy along the process.
- Your job is to create a pipeline that takes an image and modifies its style.
- The style can be established in the model or by referencing a second image.
- In the second scenario, one image is utilized for content and another for styling.
- The system should create a new image that integrates the first's content with the new style, all while ensuring that the data remains secure and encrypted.

Expectations: You are expected to demonstrate the effectiveness of your solution by providing examples and discussing tradeoffs you made to make your model compatible with FHE computation.

- To make the task manageable on common PCs, the input images can be small (for example, 32×32 or 48×48 which works well for portrait pictures).
- You may perform some pre-processing steps on clear data before running the FHE-style transfer circuit. Approaches that only perform a forward-pass of a neural network to do style transfer should be the most amenable to work with the quantization necessary to use FHE.

Your submission should contain:

- A report on the method and technical choices you made.
- A notebook showing off your model on some images

18.4 Steps in creating a privacy-preserving image style transfer application using Concrete ML and Fully Homomorphic Encryption (FHE)

- Understand the Requirements

  a) Privacy-preserving: Use FHE to ensure that the image data remains encrypted throughout the process.

  b) Style Transfer: Use neural network architectures to combine the content of one image with the style of another.

  c) Concrete ML: Leverage this framework to convert machine learning models into their homomorphic versions.

- Choose a Neural Network Architecture including;

  a) Convolutional Neural Networks (CNNs), like VGG-16 and VGG-19.

  b) Encoder-Decoder Networks.

  c) Perceptual Loss Networks, which use a loss function based on feature maps from a pre-trained network.

  d) Transformers, especially Vision Transformers (ViTs).

  e) Neural Style Transfer (NST), which optimizes an image to match content and style features from a pre-trained CNN.

Given FHE constraints, CNNs like VGG-16 or VGG-19 are typically more straightforward to adapt.

- Design the Workflow

a) Pre-processing (on clear data)

✓ Image Resizing: Scale down images to 32x32 or 48x48 to make computations manageable.

✓ Normalization: Normalize pixel values to fit within the range required by the neural network.

b) Model Conversion and Encryption

▷ Model Preparation:

✓ Select a pre-trained CNN (e.g., VGG-19).

✓ Modify the model for style transfer tasks.

- ▷ Concrete ML Integration:
- ✓ Use Concrete ML to convert the model into its homomorphic version, ensuring all computations can be performed on encrypted data.

  Style Transfer Computation
- ✓ Encrypt both content and style images using FHE.
- ✓ Perform the style transfer operation on encrypted data using the homomorphic model.
- ✓ Decrypt the output to obtain the styled image.

• Write Rust code to include; Pre-processing in Clear Data, Model Conversion and Encryption and Style Transfer Computation.

- Evaluation and Trade-offs

  a) Effectiveness: Test the model with various images to ensure the style transfer works correctly.

  b) Performance: Measure the time taken for encryption, computation, and decryption.

  ▷ Trade-offs:

  ✓ Model Complexity vs. FHE Constraints: Simplify the model if FHE computations are too slow.

  ✓ Image Size vs. Computation Time: Smaller images reduce computation time but may affect style transfer quality.

- Documentation and Reporting
  a) Method Report: Document your approach, model selection, pre-processing steps, FHE integration, and any trade-offs made.

a) Example Notebook: Create a Jupyter notebook showcasing the entire process with example images.

```
# Privacy-preserving Style Transfer using Concrete ML and FHE

## Introduction
This project demonstrates a privacy-preserving image style
 transfer application using Concrete ML to leverage Fully
  Homomorphic Encryption (FHE).

## Methodology
1. **Model Selection**: VGG-19 pre-trained on ImageNet.
2. **Pre-processing**: Images resized to 32x32 and normalized.
3. **FHE Integration**: Converted the model using Concrete ML.
4. **Style Transfer**: Performed on encrypted data.
5. **Decryption**: Final output image decrypted for visualization.

## Results
- **Example Images**: Show before and after images.
- **Performance Metrics**: Time taken for encryption,
 computation, and decryption.

## Conclusion
Discuss the trade-offs and potential improvements.
```