# Final presentation

Team03 - Parallel

OpenACC

More Science, Less Programming

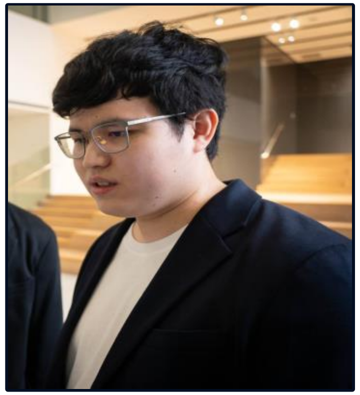
# Team03-Parallel

Mentor

**Ken Liao**

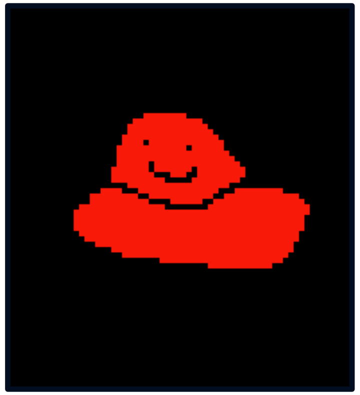
Team Members

KUAN-LIN CHEN
NCKU

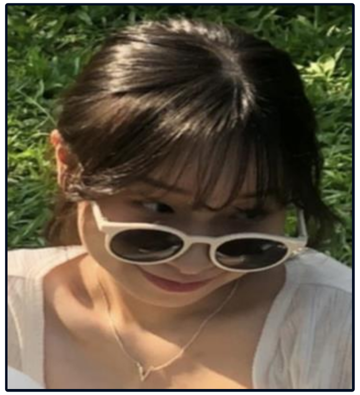


CHIA-YU YANG
NCKU

SYUAN-HAO LI
Griffith University



SHANG-CHE HSIEH
NTTU



PIN-HSUAN HO
NTTU

## Timeless Restore

- **Problem the team is trying to solve.**
  - The problem is image restoration. Specifically, the goal is to take an old, damaged, or degraded photograph (e.g., one with cracks, scratches, or color fading) and restore it to its original, clean, and high-fidelity state.
- **Scientific driver for the chosen algorithm.**
  - The scientific driver is the Denoising Diffusion Probabilistic Model .
- **What’ s the algorithmic motif?**
  - The core algorithmic motif is the U-Net architecture.
- **What parts are you focused on?**
  - The focus is on conditional guidance.
    This is not just a simple generative model; it's a restoration model.

## Evolution and Strategy

- What was your goal for coming here?
  - My goal is to learn NVIDIA 's acceleration technologies. I want to understand how to leverage the power of GPUs to significantly speed up computationally demanding tasks. To build a conditional diffusion model capable of taking a damaged photograph and restoring it to its original, clean state.

- What was your initial strategy? & How did this strategy change?

| Feature | Unconditional Generation | Conditional Restoration |
|---|---|---|
| Core Task | Generation. Create new images from pure noise. | Restoration. Repair a specific damaged image. |
| Model Input (Forward) | forward(self, x, t) | Forward (self, x, t, condition) |
| Training Objective | Predict noise from the noisy image. | Predict noise using the broken image. |

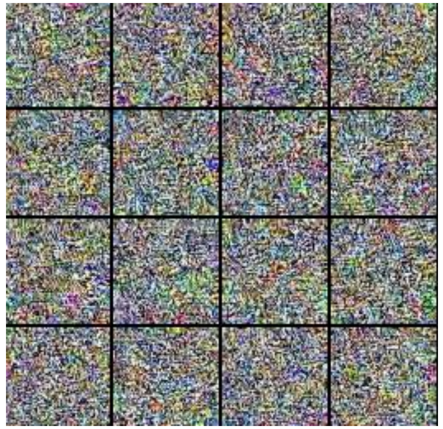


Fig.1. Unconditional Generation result.

## Dataset Generation

- Image Degradation Pipeline
  - Color Fading: Reducing saturation and applying a sepia (brownish) tone.
  - Blur & Low Contrast: Simulating blur and aged (faded) effects by reducing contrast.
  - Noise Addition: Injecting a combination of Gaussian noise and salt-and-pepper noise.
  - Scratch Generation: Creating long, straight scratches with sinusoidal variations.
  - Crack Generation: Generating branched crack masks to simulate realistic aging cracks.

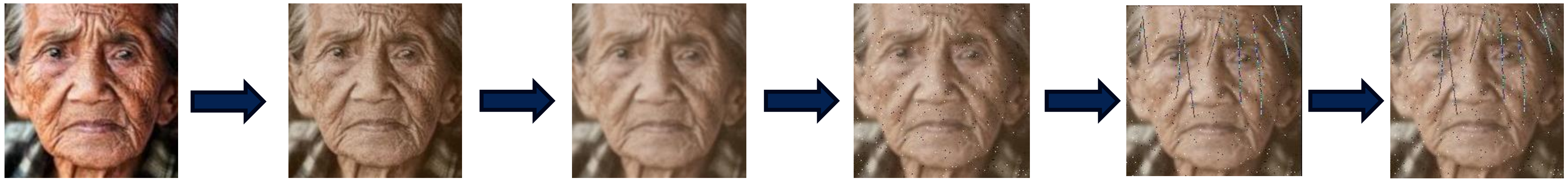
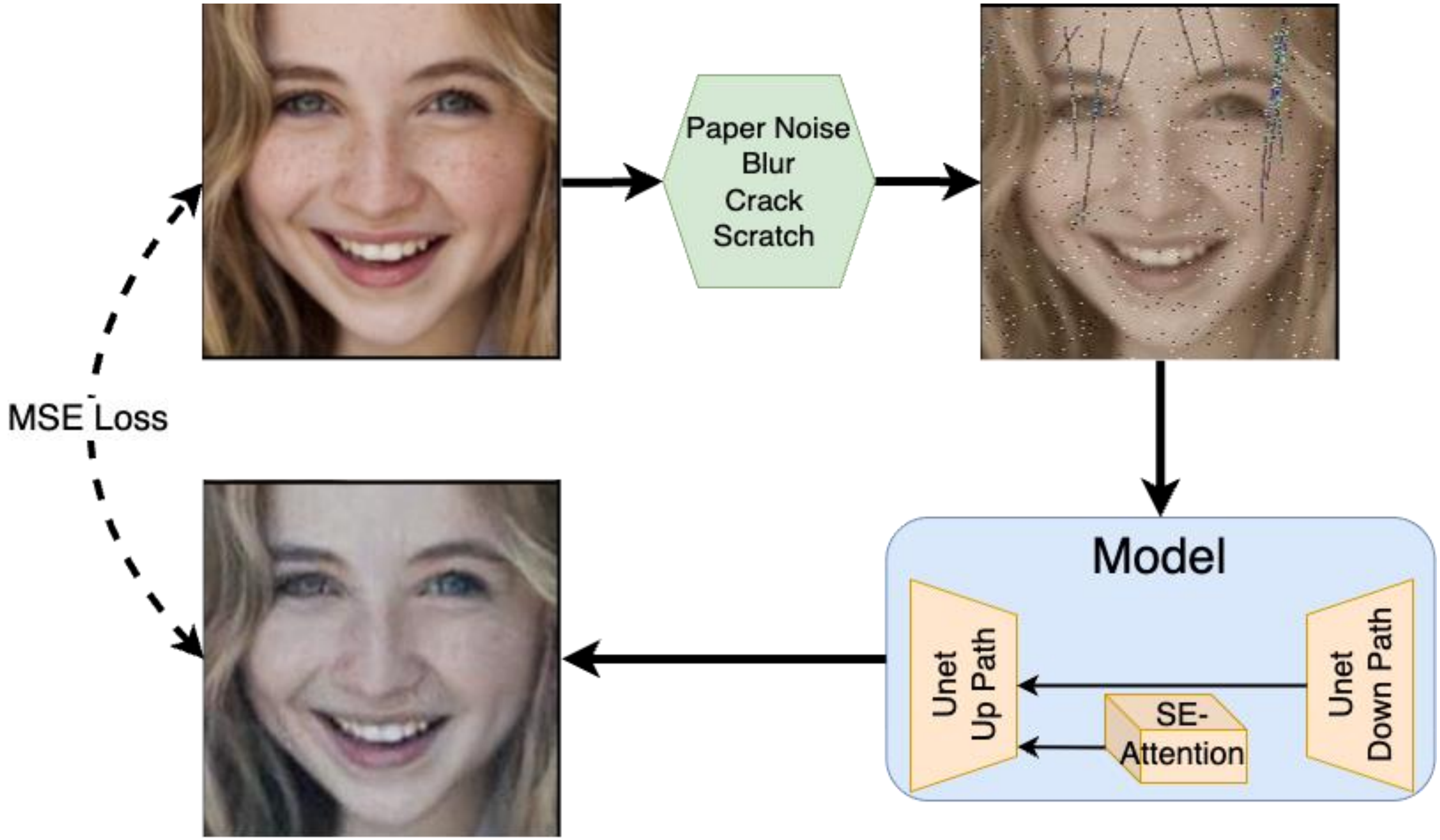
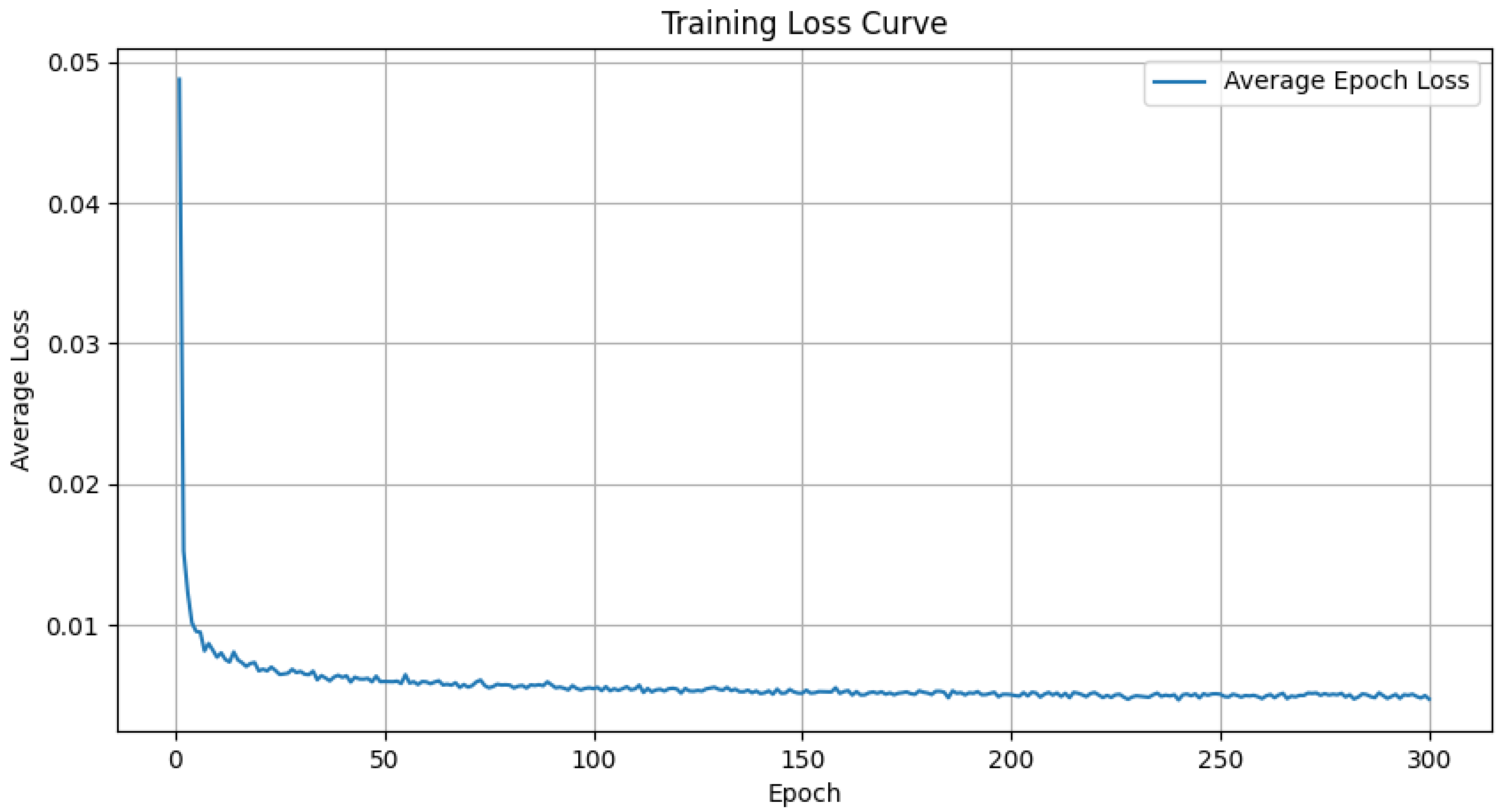


**Fig.2. The UTKFace dataset was degraded to generate paired training samples.**

## Architecture



## 300 epochs. Training: 16,595 photos. Testing: 7,113 photos.



## Output

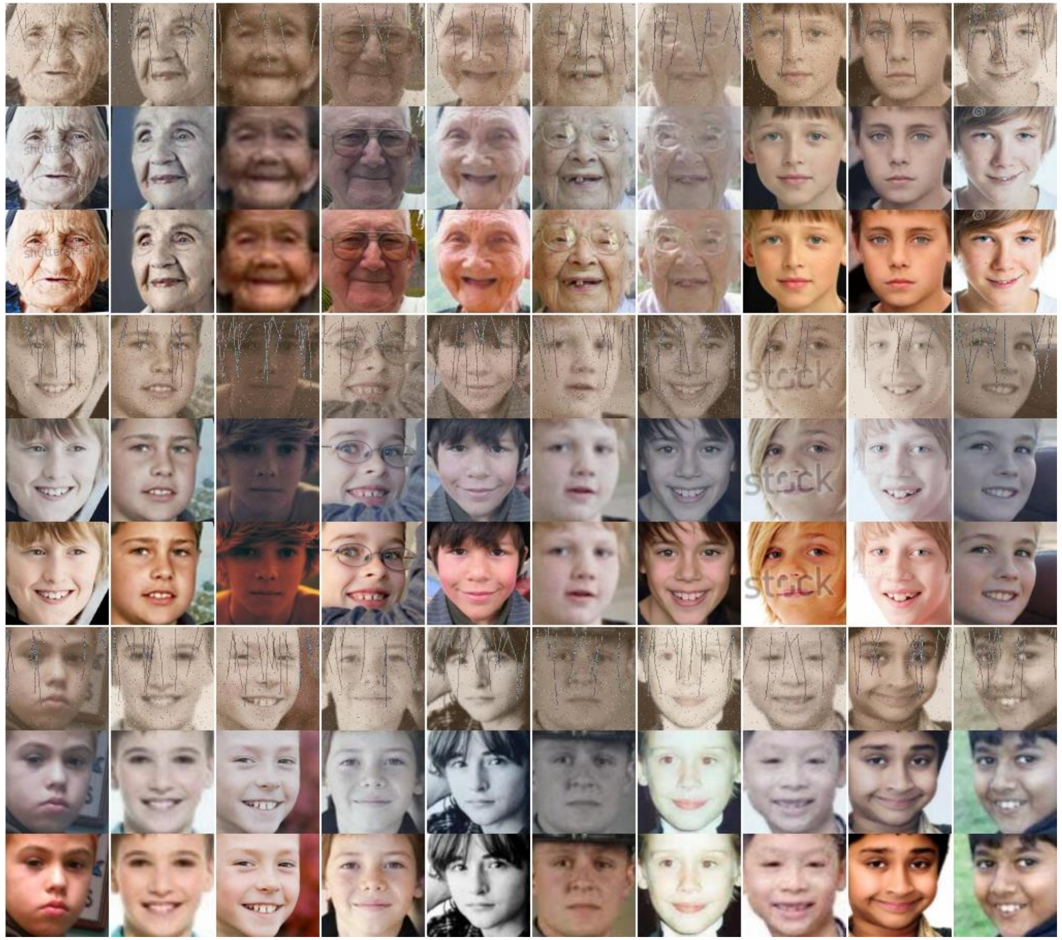
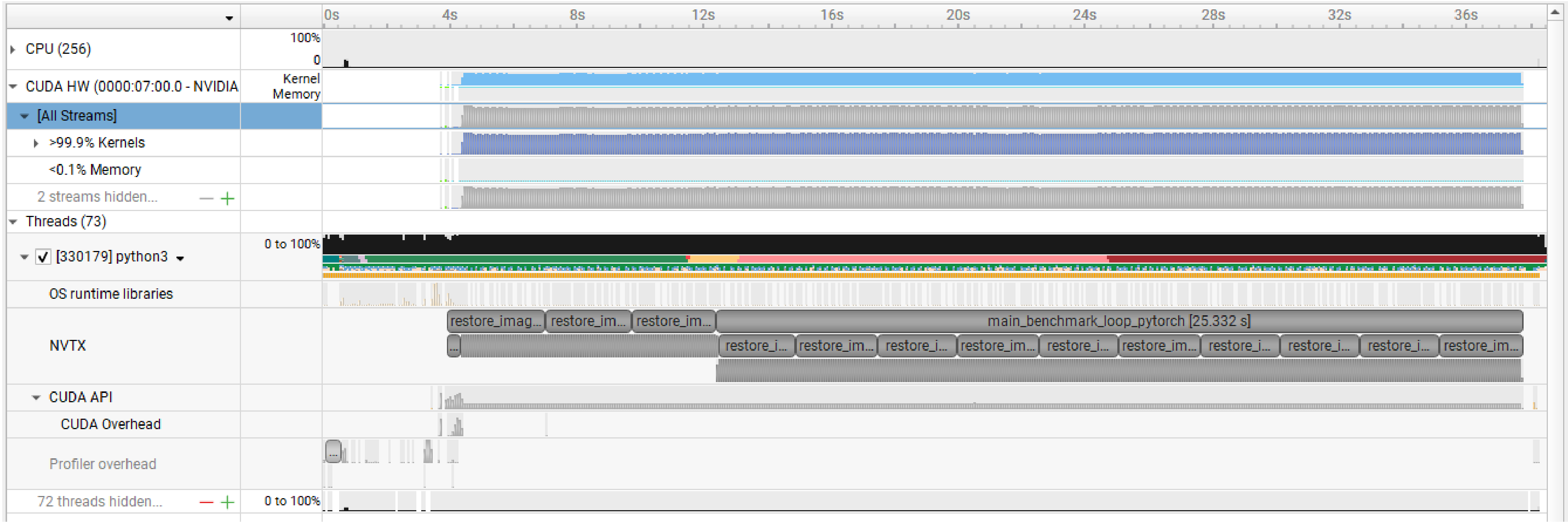
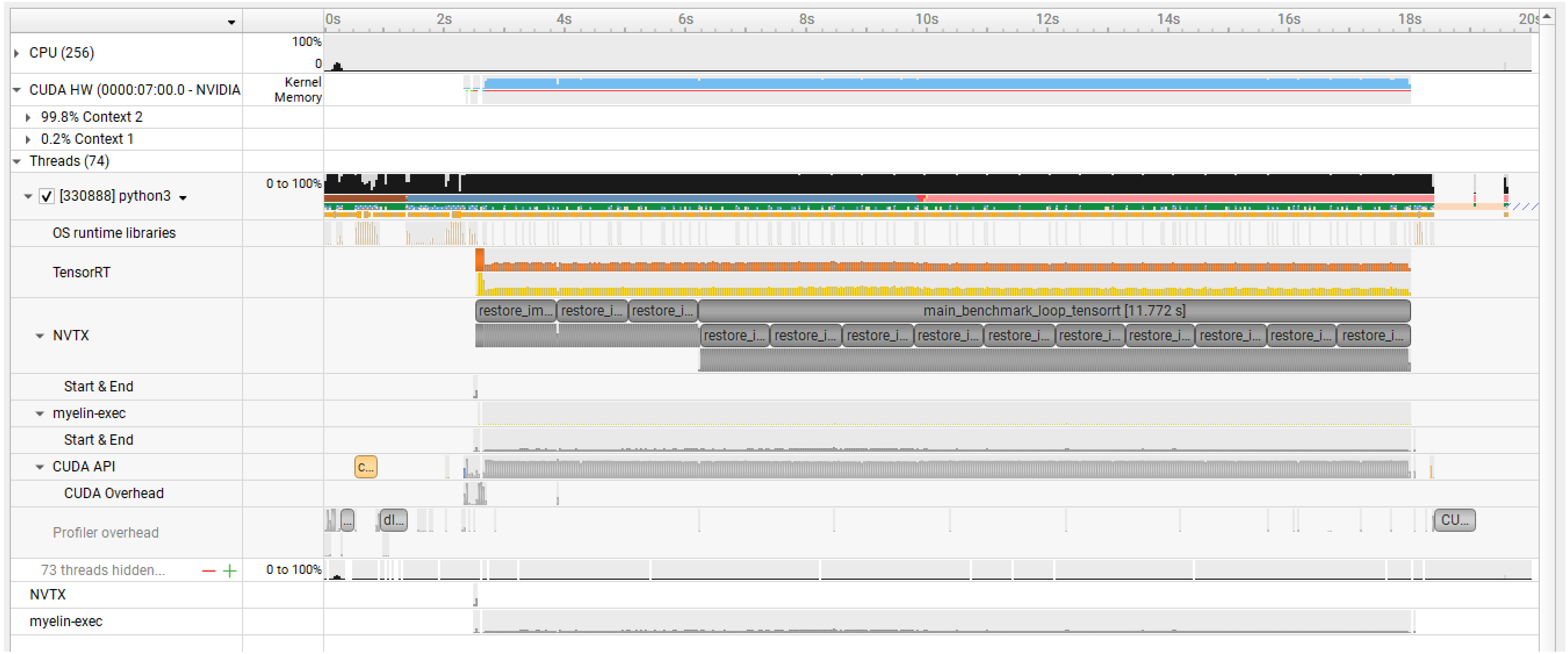


Fig.5. Qualitative restoration results on the test set.

# Restore PyTorch



# Restore TensorRT



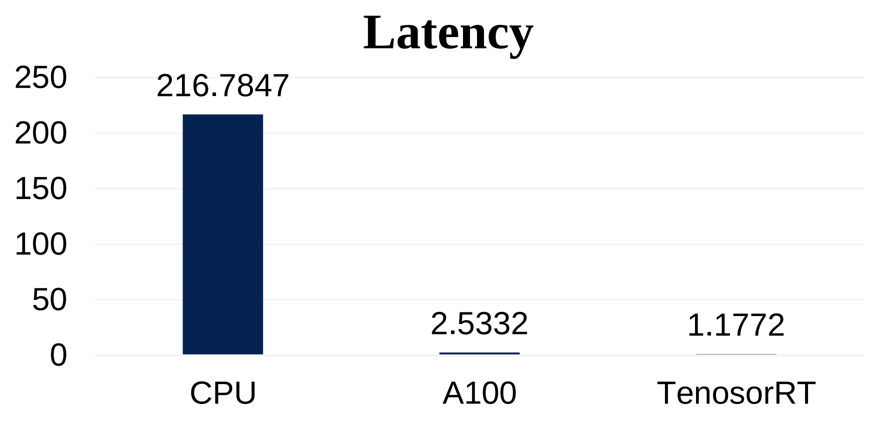
## Profiler Output

- Restore **PyTorch** Latency: **2.5332** sec/image
- Restore **CPU** Latency: **216.7847** sec/image

  Speed-up Factor = 216.7847 / 2.5332 ≈ **85.57x**

  Speed-up Factor = 216.7847 / 1.1772 ≈ 184.15x
- Restore **TensorRT** Latency: **1.1772** sec/image

  Speed-up Factor = 2.5332 / 1.1772 ≈ **2.15x**



# Energy Efficiency

| INPUTS | |
|---|---|
| # CPU Cores | 128 |
| # GPUs (A100) | 1 |
| Application Speedup | 85.0x |

| | |
|---|---|
| **Node Replacement** | 680.0x |

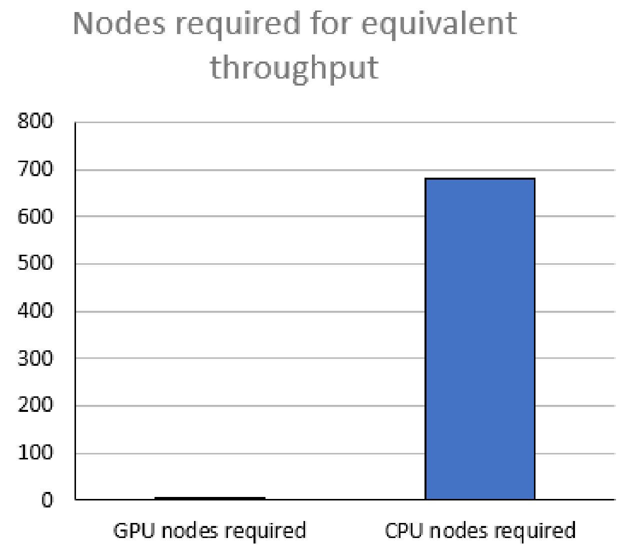
| GPU NODE POWER SAVINGS | | | |
|---|---|---|---|
| | **AMD Dual Rome 7742** | **8x A100 80GB SXM4** | **Power Savings** |
| Compute Power (W) | 748,000 | 6,500 | 741,500 |
| Networking Power (W) | 31,577 | 93 | 31,484 |
| **Total Power (W)** | **779,577** | **6,593** | 772,984 |

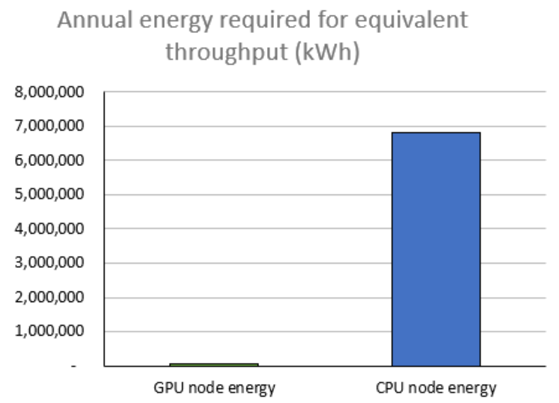
| | |
|---|---|
| **Node Power efficiency** | **118.2x** |

| ANNUAL ENERGY SAVINGS PER GPU NODE | | | |
|---|---|---|---|
| | **AMD Dual Rome 7742** | **8x A100 80GB SXM4** | **Power Savings** |
| Compute Power (kWh/year) | 6,552,480 | 56,940 | 6,495,540 |
| Networking Power (kWh/year) | 276,614 | 814 | 275,801 |
| **Total Power (kWh/year)** | **6,829,094** | **57,754** | **6,771,341** |

| | |
|---|---|
| **$/kWh** | $ 0.18 |
| **Annual Cost Savings** | $ 1,218,841.33 |
| **3-year Cost Savings** | $ 3,656,524.00 |

| | |
|---|---|
| **Metric Tons of $CO_2$** | 4,801 |
| **Gasoline Cars Driven for 1 year** | 1,036 |
| **Seedlings Trees grown for 10 years** | 79,360 |





## Final Thoughts

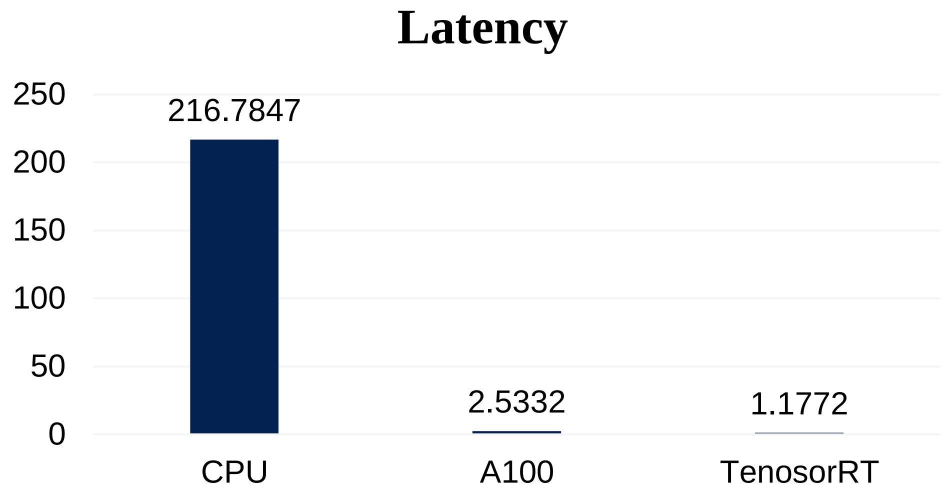
- Was this Open Hackathon worth it?
  - It was extremely valuable. We didn't just discuss theory; we achieved a concrete, measurable result.
- What did you learn?
  - We learned how to use NVIDIA tools for acceleration.
- Future Works

  Z. Wan *et al*., "Old Photo Restoration via Deep Latent Space Translation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2071-2087, 1 Feb. 2023, doi: 10.1109/TPAMI.2022.3163183.

  https://github.com/microsoft/Bringing-Old-Photos-Back-to-Life

## Application Background

Our application is an AI-powered Image Restoration System. It utilizes a Denoising Diffusion Model (DDPM) to process corrupted or degraded images, generating a clean, high-fidelity restored version.
The core computational motif is the U-Net architecture, which functions as the system's "denoiser." The inference process is inherently iterative, requiring hundreds of sequential executions of this U-Net to produce a single output image.



## Hackathon Objectives and Approach

Our primary objective was to accelerate the diffusion model's inference latency to make it deployment-ready. Our approach was to first establish a performance baseline with PyTorch, then use NVIDIA Nsight systems to profile the model, identifying the iterative U-Net execution as the main bottleneck. Finally, we converted the model into a highly optimized TensorRT engine, using its graph optimizations and layer fusion techniques to drastically reduce the latency of each step.

## Technical Accomplishments and Impact

This optimization makes the application practical. A 2.5-second latency is prohibitive for many interactive applications. By reducing the latency to ~1.17 seconds, the tool becomes responsive and viable for deployment, enabling its integration into high-performance service workflows.

# Thank You

OpenACC

More Science, Less Programming

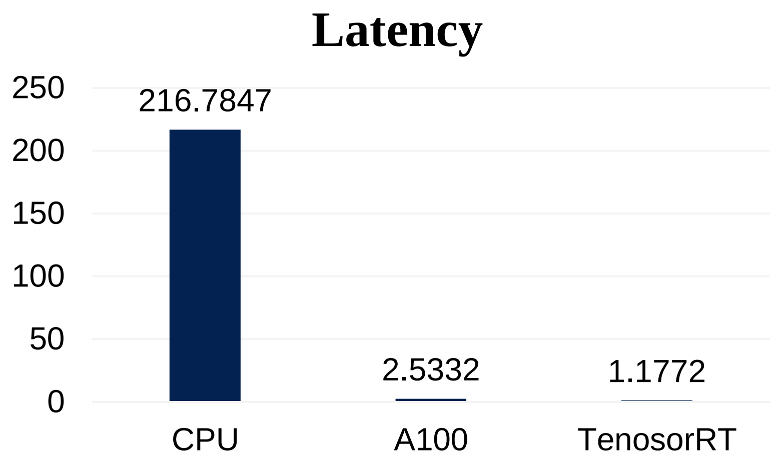
## (Required) Create a storyline for publication on NCHC's website.

AI 讓老照片秒速重生

Parallel 團隊來自成功大學，將AI 影像修復模型的推論速度加速了2.15倍！！

AI 影像修復技術能還原珍貴的老舊照片，而「擴散模型」是目前修復品質最好的技術。然而，它天生速度極慢，模型需迭代 300 次，導致推論延遲過高，難以實際應用。為了解決這個瓶頸，我們在此次駭客松中使用 NVIDIA Nsight 找出 U-Net 效能熱點，並透過 TensorRT 進行優化。我們成功將推論時間從 2.53 秒/張大幅降低至 1.17 秒/張，達成了 2.15 倍的加速。這個成果讓模型從實驗品變成了實用工具，使其能真正部署到網頁或 App 中，服務大眾。

報告投影片連結 (由國網上傳到 github)


Latency
CPU
A100
TenosorRT