*Christoph F. Eick, Raunak Sarbajna and Md/ Mahin*

COSC 4368: Fundamentals of Artificial Intelligence Spring 2024

Problem Set 3

Farzana Yasmin, Raunak Sarbajna and Christoph F. Eick

**Task 5: Creating a Neural Network and Using Diffusion Models**

Individual Task

Very Preliminary First Draft

A collage of cars

Description automatically generated

Fig. 1: The Stanford Cars dataset, compiled by Jonathan Krause

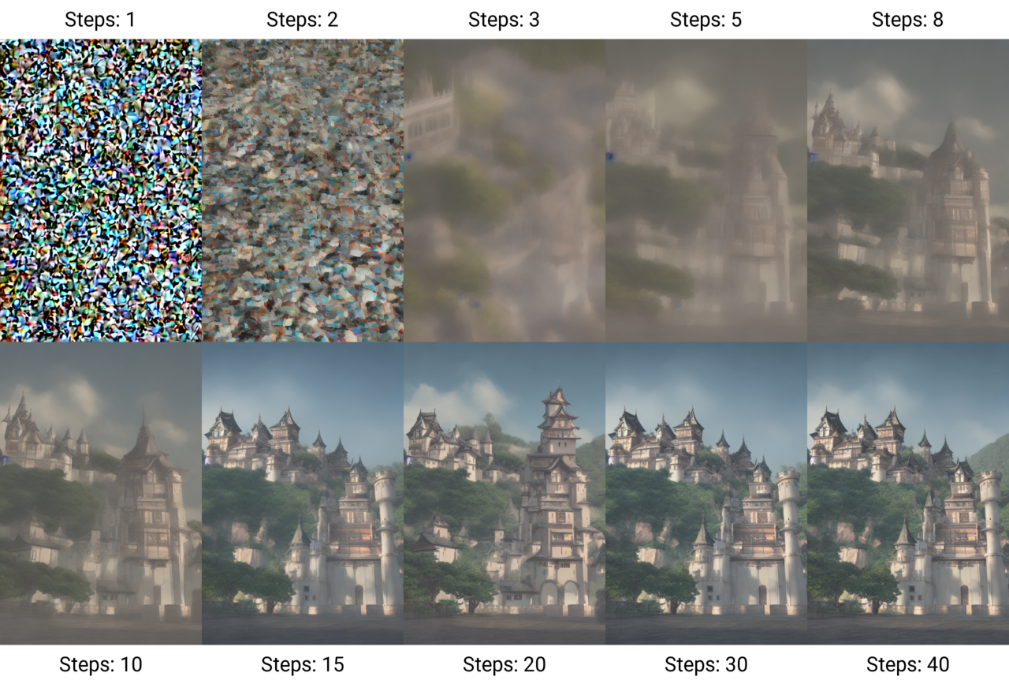


Fig. 2: The Denoising Process used by *Stable Diffusion*

Submission Deadline Task5: November 21, 11:59p

Responsible TA: Farzana Yasmin

Last Updated: September 2, 8a

Task Weight: 20% of the ProblemSet Points

This assignment is a short exploration of a how to create a neural network and a basic diffusion model. You'll be working with a pre-trained diffusion model in a Jupyter notebook environment to generate images and analyze how different settings influence the results.

Link: <https://github.com/RaunakDune/BasicDiffusionModel/>

**Learning Objectives:**

1. Understand the fundamentals of neural network architecture design.
2. Analyze model performance using various metrics.
3. Gain hands-on experience with diffusion models using a Jupyter notebook.
4. Experiment with tuning hyperparameters in a pre-trained diffusion model.
5. Analyze the generated images and interpret the impact of parameter changes.
6. Effectively document and visualize your findings within the notebook.

**Dataset**

**Link:** [**https://drive.google.com/file/d/1NF-JNSVkdu\_lG7UVVtTAUMWIRQANqtav/view?usp=sharing**](https://drive.google.com/file/d/1NF-JNSVkdu_lG7UVVtTAUMWIRQANqtav/view?usp=sharing)

The Stanford Cars dataset ([https://web.archive.org/web/20210224165646/http://ai.stanford.edu/~jkrause/cars/car\_dataset.html)](https://web.archive.org/web/20210224165646/http:/ai.stanford.edu/~jkrause/cars/car_dataset.html)%20) is a widely used benchmark dataset in the field of computer vision and machine learning, particularly for tasks related to object recognition and classification. It is curated and maintained by researchers at Stanford University.

The Stanford Cars dataset consists of images of cars belonging to 196 classes, where each class represents a different car model. The dataset is further divided into training and test sets, with a total of 16,185 images in the training set and 8,054 images in the test set. Each image in the dataset is annotated with bounding boxes around the car, along with additional metadata such as the make, model, and year of the car. This annotation facilitates fine-grained classification tasks, where the goal is to classify images into specific car models. However, we will *not* be using these annotations for this task.

**Task 5 Subtasks**

**SECTION 1**

* **Preprocess Images and Create Neural Network (20 points)**
  1. Load all images from the dataset in the link.
  2. Resize all images to 224x224 pixels and normalize all pixel values to [0, 1] range.
  3. Design and implement a CNN architecture suitable for the Stanford Cars classification task, keeping in mind the following:
     1. Use a minimum of 2 convolutional layers.
     2. Use appropriate pooling layers, include batch normalization and implement dropout for regularization.
     3. Your final dense layer *must* have 196 outputs (softmax activation).
  4. Compile and train your model.
     1. Train for 3 – 10 epochs. Let us know if you’re having trouble running the model.
     2. Implement early stopping, and save the best model based on accuracy.
* **Model Evaluation and Analysis (25 points)**
  1. Calculate accuracy, precision, recall, F1-score.
  2. Report per-class performance for 10 most/least accurate classes
  3. Show examples of correctly classified images and misclassified images
  4. Write one paragraph explaining any patterns you noted in the misclassified cars.
  5. Write one paragraph analysing which car classes are most/least distinguishable.

SECTION 2

* **Understanding the Existing Diffusion Model (No Points)**
  1. Review the provided Jupyter Notebook file containing an existing diffusion model implementation.
  2. Familiarize yourself with the structure of the code and understand how the diffusion process is simulated.
  3. Identify key parameters in the diffusion model, such as the diffusion rate, initial conditions, and time steps.
  4. Run the model once from the start to get a benchmark of the result and the time taken.
* **Implementing Parameter Tuning for Forward Diffusion (15 Points)**
  1. Explain what these parameters from the existing diffusion model do: T, IMG\_SIZE, BATCH\_SIZE.
  2. Create a new code cell in the Jupyter Notebook to modify the selected parameters.
  3. Conduct a series of simulations by varying the values of the chosen parameters while keeping other parameters constant. Example values are given below, *but you may choose your own*.
     1. T: 250/300/350
     2. IMG\_SIZE: 16/32/64/128
     3. BATCH\_SIZE: 64/128/256
  4. Record the results of each simulation, including the quality of image and time taken for diffusion.
* **Modify the Backward Process and Loss: (15 Points)**
  1. Explore the U-Net implementation and understand what it’s doing. Write a 1 to 3 line summary.
  2. Implement two different activation functions of your choice. Some suggestions are present in the notebook. Explain your choice.
  3. Explore two different loss functions of your choice in addition to the one present in code. Explain your choice.
* **Training and Experimentation (15 Points)**
  1. Choose the fastest and second fastest parameter values from **Question 2** for your training.
  2. Explore the results for each of the activation functions and loss functions.
  3. Create a variable to track the loss values and plot it.
  4. Observe how the generated cars look. Why do they look so poor? How does the quality change with IMG\_SIZE?
* **Analysis (10 Points)**
  1. Summarize your findings from all the experiments and discuss any trends or patterns observed in the results.
  2. How does each parameter affect the rate and extent of diffusion?
  3. Compare the results from the Diffusion model to your Neural Netowrk from Section 1. Are there any classes that are both hard to classify and hard to generate?

**Deliverables**

* Save your modified Jupyter Notebook with your changes and analysis.
* Ensure that your Notebook includes clear explanations, code comments, and visualizations.
* *Submit your Notebook file (.ipynb)* along with any additional resources (e.g., data files) to Teams. **Don’t zip your submission.**
* Make sure to adhere to academic integrity guidelines and cite any external sources used in your assignment.
* If you’re using an LLM (ChatGPT/Gemini/Claude/etc.) to generate your code, make sure to cite it in a comment in your code!
* Please do not use LLMs for your analysis, it is not helpful for us to grade the chatbot instead of your own understanding.

**References:**

* **Jupyter Notebook:** [**https://github.com/RaunakDune/BasicDiffusionModel/**](https://github.com/RaunakDune/BasicDiffusionModel/)
* **Dataset:** [**https://drive.google.com/file/d/1NF-JNSVkdu\_lG7UVVtTAUMWIRQANqtav/view?usp=sharing**](https://drive.google.com/file/d/1NF-JNSVkdu_lG7UVVtTAUMWIRQANqtav/view?usp=sharing)

Tong Zhou and Christoph F. Eick

Task 6: Ethical and Societal Problems of AI (Group Task)

Deadline: Monday, December 1, 11:59p

Weight: 12% of the points associated to the Problemset Tasks

Task Description will be added appox. October 31, 2025