Exploring image generation with unconditional DDPM and conditional DDPM with classifier-free guidance

Anonymous Full Paper
Submission 42

Abstract

This project explored image generation with DDPM models based on the open-source Oxford-flowers dataset. An unconditional DDPM and a conditional DDPM with classifier-free guidance were implemented to explore how changes to the model specification and hyperparameters affect image generation performance. The project analysed how increasing the training batch size, applying different image augmentations and changes to the U-Net model architecture affect the quality of the generated images and inference time. The generated images were analyzed both visually and by calculating three different no-reference image quality metrics. It was concluded based on visual inspection and quality metric scores that larger batch size improves image quality and image augmentations tested in the scope of this project lower the quality of the generated images. In addition, a large improvement in the quality of the generated images was seen based on visual inspection when a more complex three-layer U-Net model architecture with attention layers was applied, compared to the simpler two-layer U-Net architecture. Lastly, the project work explored how the guidance scale value in the classifier-free guidance model affects image generation. Fréchet inception distances (FID) and Inception Scores (IS) were computed for imaged generated with five different guidance scales and used to compare the trade-off between the adherence to the conditioning signal and diversity in the generated images.

1 Introduction

Generative models have revolutionized how we create and interpret visual content. Recent advancements in the generative modelling field have enabled the synthesis of detailed realistic high-quality images [1–3] and the development of text-to-image methods has enabled the image generation process to be guided in detail through text prompts [4, 5]. Release of open-source text-to-image generation model DALL-E2 [2], and similar others [1, 3, 6], and the recent announcement of text-to-video generation model Sora [7] have made generative models easily accessible and opened up new possibilities for various industries. Generative models have been applied in the entertainment industry to create environments or characters for movies and video games [8, 9], in advertising to produce promotional materials [10] and in medical imaging to improve the quality of diagnostic images [11].

Deep learning is at the heart of the image generation task. Generative models can learn the different patterns and characteristics of the training data and are the starting point for creating new images [12]. While unconditional image generation relies on deep learning models that can learn the features of the training images, text-to-image generation can in addition involve Natural Language Processing (NLP) techniques that also apply deep learning approaches to model human language [13].

Among others, denoising diffusion probabilistic models have emerged as a powerful new tool for image synthesis. They first “destroy” the training data by adding noise to the training images (forward diffusion process) and then apply deep learning approaches to reverse this noising process (reverse denoising process) [14]. After training, the model can be used to generate new images by passing randomly sampled noise through the deep learning model [15]. Similarly to other generative models, DDPMs performance and the quality of generated images directly depend on the architecture of the underlying deep learning models, how these models are trained and the different hyperparameters associated with model training and the inference process [16].

The aim of this project was to explore the task of image generation based on DDPMs. Both unconditional and conditional image generation models were implemented and compared. The main aim of this project was to explore how image generation performance and inference time are affected by different data transformations and changes to the model architecture and training process. The generated images were evaluated both visually and based on no-reference image quality metrics, to see if the metrics and visual inspection are in agreement. Lastly, the project work explored how the guidance amount in the classifier-free DDPM affects image generation. Images generated with different guidance amounts were compared both visually and based on the FID and IS values.

2 Literature Review

2.1 Background

Deep learning is a branch of machine learning that is capable of learning from data by passing it through a multi-layer network. For the task of image generation, the input data is images that are represented as two or three-dimensional arrays of pixel values. To train a deep learning model from scratch, often a large amount of data is needed to achieve good performance of the model [17, 18]. This is also the reason why deep learning algorithms often require larger computational resources compared to other standard machine learning models [19]. To increase the data size and avoid overfitting, data augmentation is often applied to the training data [20]. For images, this can include horizontal and vertical shifts and flips, small rotations, scaling or colour adjustments [20].

Deep learning methods originate from artificial neural networks, which are composed of multiple connected layers [21]. Processing elements called artificial neurons are the building blocks of the model layers. Each neuron takes the input data and the parameter weights and creates a real-valued activation as an output [22]. Each information-processing layer consists of a number of neurons and the layers are structured in a hierarchical manner [22]. The input data is first passed to the input layer consisting of a set of units. Each unit in the input layer then feeds information to all neurons in the first hidden layer, where activations are calculated based on the input and a prespecified nonlinear activation function. Hidden layers pass the information on to the next layer until the output layer is reached, which returns the result. [23] In the task of image generation, the output will be a two- or three-dimensional array which represents the generated image.

An extension of the feed-forward networks described above is the regularized type of neural network called convolutional neural network (CNN) which learns local features or patterns in the images by combining convolution and pooling hidden layers [24]. The motivation behind CNNs is that nearby pixels are more strongly correlated than pixels far away from each other. Instead of fully connected layers, CNNs contain convolution layers, which capture the small details in the image and pooling layers, which downsample the convoluted image to select the subset of features that are informative [23]. The convolution layers contain filters that perform convolutions, which entails taking the dot product of the convolution filter and the input. The output of the convolution layer is two-dimensional feature maps, where regions in the image that resemble the given convolution filters will be highlighted [23]. Usually, Rectified Linear Unit (ReLU) activation is applied to the convolution layer output before passing it on to the next layer [25]. Pooling layers that follow the convolutions will condense the input to smaller dimensions by combining the information from several pixels. Often maximum or average pooling is used [26]. The total architecture of CNNs usually consists of several convolution and pooling layers followed by one or more fully connected layers.

Before deep learning models can be applied to the task at hand, the unknown parameters (weights) of the model need to be estimated. Deep learning models are fitted in an iterative fashion using stochastic gradient descent [27]. The training starts with a random guess for the model weights and after each iteration model weights are updated in a way that would minimize the predefined loss function [23]. Backpropagation is used to find the parameter values that minimize the loss [28] and the learning rate parameter controls the step size when moving towards the minimum of the loss function [29]. Training is carried out until the loss stops decreasing. To avoid model overfitting, several approaches for regularization are often applied, such as adding a penalty term to the loss function [30] or dropout, which sets the activations of a small number of random units to zero during the processing of each training observation [31].

2.2 Diffusion models

The idea of generative diffusion models was first introduced in 2015 and was originally derived from non-equilibrium statistical physics [14]. In diffusion models, the structure of the data is first iteratively destroyed by adding Gaussian noise and then the reverse diffusion process is learned that restores the data structure [14]. Since the publication of Song *et al.* in 2019 [32] and Ho *et al.* in 2020 [15], who independently improved the diffusion model idea, these models have gained a lot of popularity. Research and application of these models have exploded in recent years and resulted in generative models that can produce state-of-the-art image quality [2, 6, 16].

Several benefits of diffusion models make them attractive for image generation. The diffusion model fitting objective is to maximize the likelihood in comparison to Generative Adversarial Networks (GAN), where adversarial training is needed [15]. Therefore, diffusion models can cover the total probability distribution of the training data set and can generate images with high diversity [16]. Also, unlike Variational Autoencoders (VAEs) which generate images at once from the latent space [33], diffusion models gradually remove noise during image generation [15]. By first creating a rough representation of the image structure and then adding details to the generated image, diffusion models can create images with high fidelity [34]. Lastly, diffusion models are well scalable and parallelizable [35, 36]. These advantages make diffusion models attractive for several applications. They are applied in computer vision domain for tasks such as image and video generation [2, 37], inpainting [38], restoration [39], and editing [40], but have also shown good performance for temporal data modelling tasks [41], computational chemistry [42] and medical image reconstruction [43].

DDPMs consist of two parts that are both carried out for a finite number of time steps: a forward diffusion process generates noise from an input image and the reverse denoising diffusion process turns noise into a new image [15]. In the forward diffusion process, q, the training images are mapped to the latent space using a Markov Chain. This can be done by sampling noise from a Gaussian distribution at each time step based on a variance schedule and adding this noise to the image of the previous time step. The variance schedule is predefined and often linear or cosine schedule is applied [44]. However, as we sample from a Gaussian distribution at each time step and the sum of Gaussians is Gaussian [45], we do not need to sample the forward process repeatedly but can obtain the noisy image conditioned on the input image. The forward diffusion process results in pure Gaussian noise that has the same dimensionality as the input image [15].

The reverse denoising diffusion process, p, is learned by training a neural network to gradually denoise an image [15]. The neural network aims to approximate the conditional distribution of the image in the previous time step, given the noisy image, $p(x_{t-1}|x_t)$. Assuming that the reverse process is Gaussian, the network needs to learn the mean and the variance of this distribution. However, if the variance of this Gaussian distribution is kept fixed, the network only needs to learn the mean of the conditional probability distribution. A nice property proposed by Ho *et al.* 2020 [15] is that the mean of the Gaussian distribution can be reparametrized to make the neural network predict the added noise instead of the mean of the conditional Gaussian distribution. As the forward noising process is fixed, we can therefore optimize the model by using a simple mean squared error (MSE) between the true noise sampled in the forward process and the predicted Gaussian noise [15].

To minimize the loss function the neural network needs to take a noisy image and predict the added noise, which will be an array with the same dimensionality as the input image [15]. Therefore, as input and output have the same dimensionality, U-Net-like architecture is most often applied for training diffusion models. U-Nets are convolutional networks that were proposed in 2015 for image segmentation [46] and build upon the architecture of the fully convolutional network [47]. U-Net architecture consists of a contracting path and an expansive path. The contracting part of the network contains encoder layers that reduce the resolution of the input and follows the typical architecture of a CNN [46]. The expansive path contains decoder layers that upsample the decoded feature maps while performing convolutions. In the expansive path, information from the contracting path can be used by skip connections [48]. This is done to preserve the spatial information that was removed in the contracting path and locate the features of the encoded feature maps accurately on the final output that will be the same size as the input image [46]. In addition, as the parameters of the U-Net are shared across time points, the DDPM model employs an embedding to encode the time points [15]. This makes sure that the neural network is aware of the time step that it is currently operating on. DDPM authors also apply attention layers between the convolutional blocks [15]. Attention layers originate from the Transformer architecture and weigh the encoded data, which enables the decoder to focus the informative parts of the image flexibly [49].

After the publication of the DDPM model described above, research and development of diffusion models has increased and many improvements and alternative solutions have been proposed that can improve image quality or inference time. For example, it has been shown that learning the variance of the conditional Gaussian distribution for the denoising process can improve model performance and the cosine variance schedule can also improve results [50]. In addition, a cascaded diffusion model has been proposed that uses a pipeline of multiple diffusion models and improves image fidelity [35]. Classifier guidance was proposed to improve conditional diffusion models' image quality by using an separate trained classifier [16]. This work was further extended by classifier-free diffusion guidance that trains jointly a conditional and unconditional diffusion model with a single network [34]. One additional recent advancement is latent diffusion models, which implement the diffusion on a compressed image representation instead of the image itself, which greatly reduces computational requirements while retaining the state-of-the-art image quality [6]. A second recent development that has gained popularity is CLIP latents that turn the input text caption into a CLIP image embedding, which results in more diverse images [2].

3 Results

This project work explored the task of image generation with DDPMs. Even though a lot of powerful models and recent advancements have been proposed, the DDPM models tested here were constrained to simpler implementations that are similar to the original implementation of the DDPM [15] and classifier-free guidance for DDPMs [34]. The code for the project work is available on github [51]. The model implementations were based on three repositories [52–54], but extended to work with higher resolution images, RGB images and modified in regards to other details as needed. All models were implemented in PyTorch and run on one NVIDIA Tesla V100 GPU with CUDA 12.2.

Three no-reference image quality metrics were chosen to evaluate the generated images in addition to visual inspection: CNN-IQA, NIQE and BRISQUE. No-reference metrics were chosen as the generated images have no specific reference images. The three quality metrics were chosen as they are well-known and do not require extensive resources and are therefore easy to integrate to the inference workflow. CNN-IQA [55] applies a CNN to estimate the quality and takes patches of images as input. For CNN-IQA, a higher score indicates better image quality. Natural image quality evaluator (NIQE) metric [56] calculates image quality based on deviations from statistical regularities observed in natural images. A lower NIQE score indicates better quality. BRISQUE score is based on extracting Natural Scene Statistics and calculating the feature vectors based on the statistics [57]. A lower BRISQUE score indicates better image quality. These three metrics were calculated to see if their values agree with the visual inspection of the images.

The data set used for this work is a reduced version of the Oxford-Flowers data set [58]. The original data set was manually filtered to contain images where only one flower is present on the image, the background is uniform and does not contain additional objects, the flower is centered on the image and makes up the majority of the image space. The reduced version of the data set contained 2450 images from 61 classes and can be accessed through the HuggingFace website [59]. Before the data set was applied for model training, classes that had less than 20 images were removed. Therefore, the final data set applied for training contained 2346 images from 53 classes. As the images had different sizes, the images were center-cropped based on the shorter edge length to equal the height and width and then resized to dimensionality of 128x128. As in the original DDPM implementation, random horizontal flips were applied to the training set and the pixel values were normalized.

First, an unconditional DDPM model and a conditional DDPM model were implemented. The latter was implemented as a DDPM model with classifier-free guidance. The guidance scale w was initially set to 0 during the inference, so the model behaved as a conditional model, and class labels were dropped with a probability of 0.1. In the first models, a two-layer U-Net model architecture was applied. Time and context for the conditional model were embedded and concatenated with the feature maps. A linear beta schedule between $1 * 10^{-4}$ and $2 * 10^{-2}$ was used for variances. The batch size for model training was 16, the model was trained for 80 epochs, the initial learning rate of $1 * 10^{-4}$ was applied with linear decay, and samples of generated images were obtained with 800 time steps.

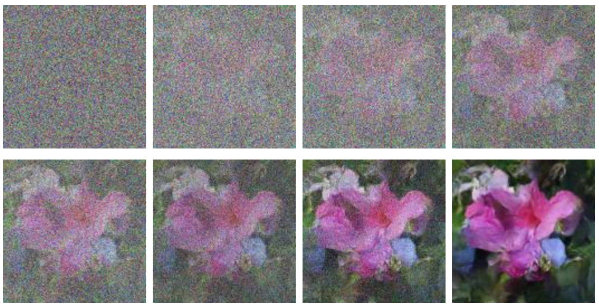
Figure 1. Images generated during the denoising process at decreasing time steps.

Examples of images generated at different time steps during the denoising process with the unconditional model can be seen in Figure 1. From the images in Figure 1, it can be seen that as the time step decreases, noise is gradually removed from the image, as expected. First, the general shape of the flower blossoms becomes distinguishable from the noise and as the time step decreases, details are added to the image. A set of images generated with the unconditional model and the conditional model with 800 time steps can be seen in Figure 2. Visual inspection of the generated images shows that both models generate colourful patches that resemble flowers on a green background. However, the details of the flower blossoms are not distinguishable. When analyzing all 106 generated images visually from both models, they seemed to perform equally well and neither stood out by generating images of higher quality. The no-reference quality metrics of the generated images for the unconditional and conditional models can be seen in Table 1. The CNN-IQA and NIQE metrics reflect the visual inspection well and are similar for the unconditional and conditional two-layer U-Net models. The BRISQUE metric is lower and indicates higher image quality for the unconditional two-layer model.

Next, to test whether using a larger batch size has an effect on the inference results, a batch size was increased from 16 to 32 for both the conditional and unconditional model. Rest of the model architecture and hyperparameters were unchanged. Calculated mean quality metric values can be seen in Table 1. An improvement in image quality was seen in all metrics for both models compared to using batch size of 16, which aligned with the visual inspection of the generated images. It should be noted that the BRISQUE score of the images generated with the unconditional DDPM showed a remarkable improvement, even though as large of an improvement

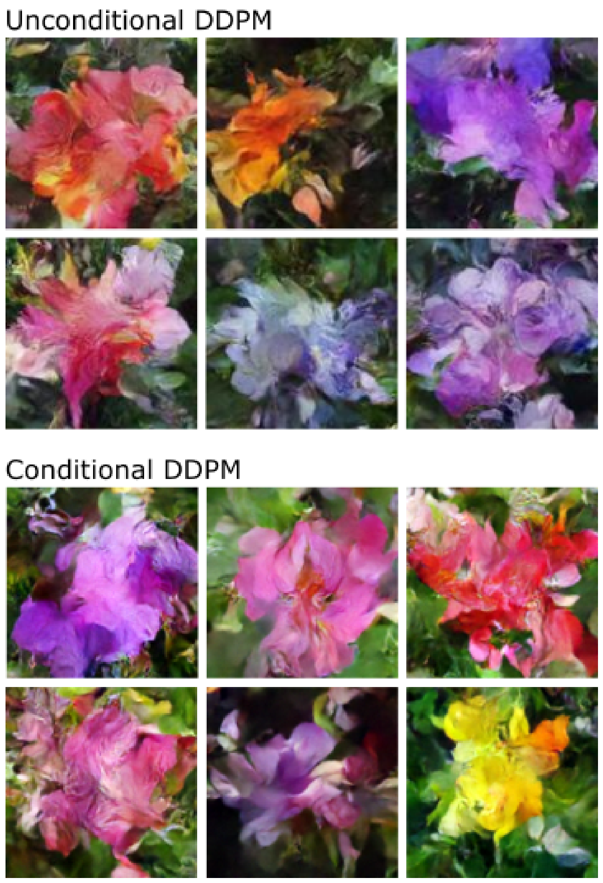

Figure 2. Images generated after 800 time steps with the two-layer U-Net architecture unconditional DDPM (upper panels) and conditional DDPM (lower panels).

Table 1. No-reference quality metrics values for images generated with unconditional DDPM models (top) and conditional DDPM models (bottom). All quality metrics are reported as an average metric value of 106 generated images. 2-layer U-Net: U-Net model architecture with two decoder and encoder layers; 3-layer U-Net: U-Net model architecture with three decoder and encoder layers; img aug.: additional image augmentations applied to input data; b.s.32: batch size of 32 (instead of 16). + indicates that the new feature was added to the model specification of the previous row.

unconditional DDPM model	CNN-IQA, NIQE, BRISQUE
2-layer U-Net	0.49, 4.50, 11.81
+ b.s.32	0.50, 4.38, 9.54
+ img. aug	0.54, 21.53, 66.21
3-layer U-Net	0.52, 4.95, 12.45
conditional DDPM model	**CNN-IQA, NIQE, BRISQUE**
2-layer U-Net	0.48, 4.90, 14.51
+ b.s.32	0.48, 4.78, 13.38
+ img. aug	0.15, 17.89, 50.01
3-layer U-Net	0.54, 4.97, 17.20

was not observed visually. Overall, the visual comparison of the generated images indicated that the conditional model generated more images where the flower blossom and the background of the image were well separated. However, this was not reflected in the quality metrics, as all metrics indicated that the images generated with the unconditional model were of higher quality. Examples of images generated with the unconditional and conditional model when the batch size was set to 32 during training can be seen in Figure 3 In general, the examples on Figure 3 show that while there is a better separation of the background and the flower object, the images do not show clear flower blossom shapes and details.

Another test was carried out based on the preprocessing of the training data. In addition to the random horizontal flips applied to the training images, random adjustments were also applied to the image brightness, contrast, saturation, hue and sharpness. All adjustments except sharpness were applied with the PyTorch ColorJitter function with 0.5 scale. The sharpness was adjusted with the RandomAdjustSharpness function with the sharpness factor of 1.5. The batch size was kept at 32 for this analysis and denoising process was carried out for 800 time steps as before. Visual inspection of the generated images after applying the aforementioned image augmentations shows that the images are of very low quality and no clear flower features can be distinguished from the noise. This might have been caused by several reasons. The image augmentations might have reduced the model capability of learning the features of the images and made it difficult to differentiate between the flowers and the background of the images. It is also possible that more time would have been needed for denoising the images to improve the results. Image quality metrics calculated after applying these image augmentations can be seen in Table 1. The NIQE and BRISQUE metric values are remarkably high for both the unconditional and conditional model compared to the values of previous tests and this well aligns with the visual inspection. However, the CNN-IQA metric shows contrasting results. The high CNN-IQA value of 0.54 for the images generated with the unconditional model indicates that the images were of higher quality compared to previously generated images where no additional image augmentations were applied. This does not align with the visual inspection. The CNN-IQA value for the conditional model is considerably lower than before, indicating that the images are of lower quality compared to previously generated images, which aligns well with the visual inspection. This contradiction of CNN-IQA metric values is unexpected and visual inspection of the images does not propose a straightforward explanation for this difference. Nevertheless, the contradicting results of the CNN-IQA metric pro-

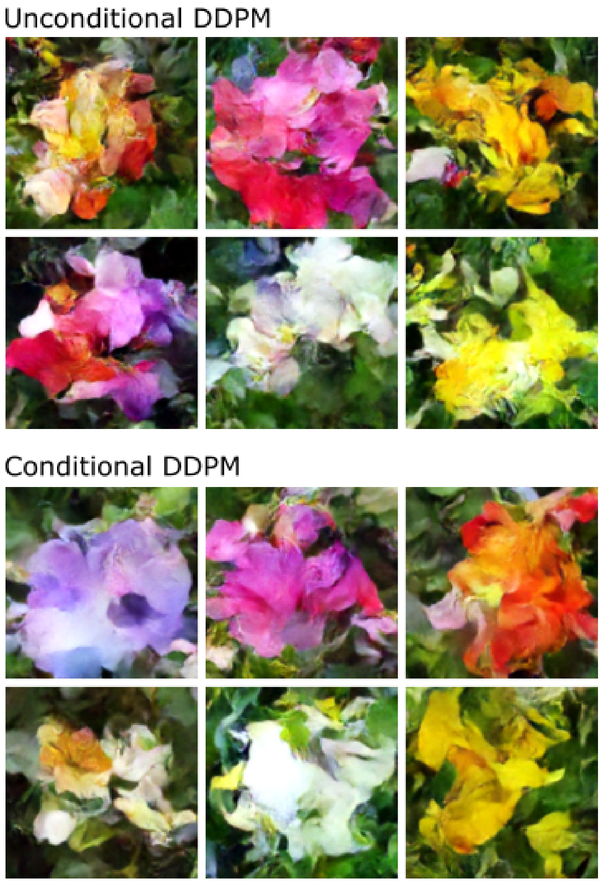

Figure 3. Images generated after 800 time steps with the 2-layer U-Net architecture unconditional DDPM (upper panels) and conditional DDPM (lower panels) when batch size was set to 32 during model training.

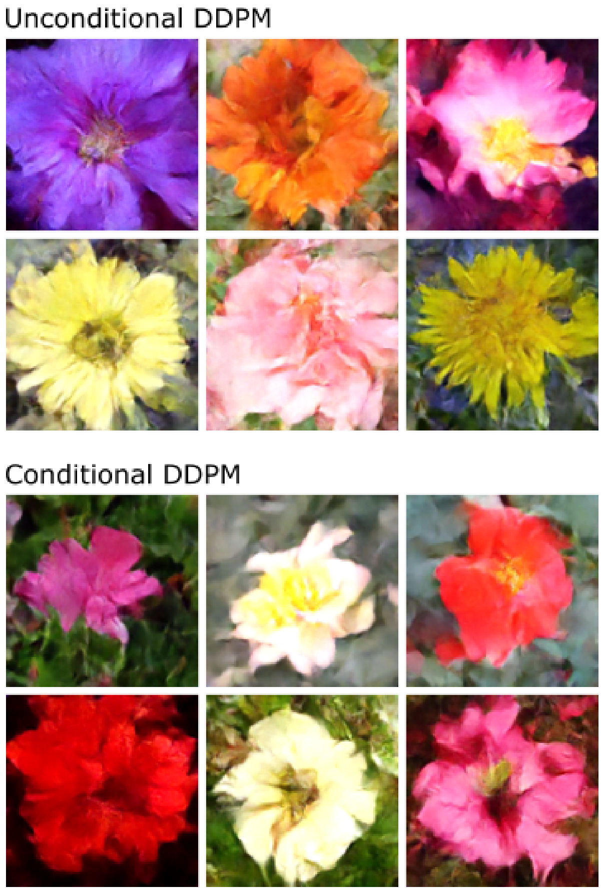

Figure 4. Images generated after 800 time steps with the 3-layer U-Net architecture unconditional DDPM (upper panels) and conditional DDPM (lower panels).

vide a good example of the importance of visual inspection of the generated images and support the idea of calculating several quality metrics for image evaluation.

Lastly, a model with a more complex architecture was tested. A new U-Net model with three encoder and decoder layers was implemented, which also included additional attention layers. The hypothesis here was that attention layers and additional convolutions can improve the model inference, as the larger U-Net model should be able to learn more complicated features of the images. The same three-layer model was applied in both unconditional and conditional DDPM. The batch size was kept at 16 for training this model to keep the memory requirements low and no additional image augmentations were applied besides random horizontal flips. Visual evaluation of the generated images showed a clear improvement in the image quality. Both the general shape of the flowers as well as specific details were more clearly visualized. Examples of images generated with the three-layer U-Net models can be seen in Figure 4.

The quality metrics calculated for the images generated with the three-layer U-Net models can be seen in Table 1. Although a remarkable improvement of image quality was observed, this was not clearly reflected in the metric values. Only the CNN-IQA metric for the conditional model showed an improvement compared to the two-layer U-Net models. Nevertheless, the NIQE and BRISQUE scores were not remarkably higher compared to the two-layer model that was trained with the batch size of 32 for neither the unconditional or conditional model. Small differences in the metric values might be explained by the relatively small number of images (106 images), that was generated and used to calculate the mean quality scores.

The training times and inference times for all models that were trained and used to generate images can be seen in Table 2. The training times of the two-layer unconditional and conditional models were similar. Extending the model architecture to the three-layer model made the training time approximately four times longer compared to the two-layer U-Net models. The inference times varied across the unconditional and conditional model implementations. This was also expected, as the conditional model inference required calculating the loss for both the unconditional and conditional instance and this

Table 2. Model training and inference times. The times are reported in the format of hours:minutes:seconds. Training time was calculated as the total training time of 80 epochs. Inference time was calculated based on the generation time of 106 images with 800 time steps. 2-layer U-Net: U-Net model architecture with two decoder and encoder layers; 3-layer U-Net: U-Net model architecture with three decoder and encoder layers; img aug.: additional image augmentations applied to input data; b.s.32: batch size of 32 (instead of 16). + indicates that the new feature was added to the model specification of the previous row.

unconditional DDPM model	Training time	Inference time
2-layer U-Net	0:32:46	0:05:06
+ b.s.32	0:31:05	0:04:58
+ img aug.	0:30:53	0:04:59
3-layer U-Net	2:11:53	0:20:12
conditional DDPM model	**Training time**	**Inference time**
2-layer U-Net	0:32:47	0:10:05
+ b.s.32	0:30:54	0:10:10
+ img aug.	0:30:52	0:10:08
3-layer U-Net	2:11:25	0:25:24

was done by doubling the data size. In general, implementations of the DDPM models presented here were resource efficient, which was important for carrying out a number of tests described above.

As the last analysis, inference of the classifier-free guidance model with the three-layer U-Net architecture was explored by testing different guidance scales. Five guidance scales were selected in addition to the guidance scale of zero which was applied in the previous inference tests. Each guidance scale value was used to generate 53 images, which were in turn used to calculate the FID [60] and IS scores [61]. The results can be seen in Table 3. Smaller FID scores indicate that the images are similar to the reference set and higher IS values indicate that the images are distinctly identifiable based on their label. In general, we expect that as the guidance scale value increases, the images are more guided based on the label, the diversity of the images is reduced and therefore the metric values increase [34]. The values reported here are not close to the optimal values that generally are reported and this is due to several reasons. First, the models applied for this project work were kept to relatively minimal implementations to manage the limited available computational resources. More advanced modeling and variance scheduling could improve the FID and IS scores. Second, the FID and IS scores are usually calculated based on image sets that are several orders of magnitude larger. Usually 10 000 to 50 000 images are generated and size of the image sets

Table 3. FID and IS scores for images generated with the DDPM model with classifier-free guidance, w: guidance scale.

w	FID	IS
0	317.29	1.87
0.1	309.76	1.80
0.5	330.88	1.75
1	320.70	1.74
2	310.38	1.88
4	331.78	2.10

used for the calculation can affect the obtained values [62]. Generating tens of thousands of images was not possible for this project work due to limited computational resources. Still, the expectation for this minimal test was that when comparing the FID and IS values across the guidance scales, the general trend of the metrics increasing will be observed. The results in Table 3 show that the increasing trend was not observed and metric values vary across the guidance scales. Nevertheless, the expected trade-off can be observed when considering the lowest FID and highest IS values. Best FID score is generally observed for a small guidance scale and the optimal IS value for a large guidance scale. This was also observed here. Even though the differences in the metric values are small, the lowest FID score was obtained with the guidance scale of 0.1 and the highest IS value was observed for the guidance scale of 4.

4 Discussion

This work explored image generation based the Oxford-Flowers data set and on two relatively simple implementations of the DDPM model. The choice of implementing the models from scratch was made to thoroughly understand the implementation of both the unconditional DDPM and classifier-free guidance model. This also set some boundaries to the quality of the generated images. There is a number more extensive and powerful implementations of DDPM models available. One good example is the Python Diffusers library, which includes a large number of pretrained models, variance schedulers and pipelines for inference which can be fine-tuned and combined for the task at hand [63].

Nevertheless, the simple implementations used in this project allowed for efficient training and inference times and low memory requirements, which allowed to carry out a number of tests based on batch size, image augmentations and two different U-Net model architectures. As models trained with the batch size of 32 showed improved performance, additional experiments with larger batch sizes could have been carried out. However, this would have also required larger computational resources. Still, implementing the 3-layer U-Net architecture considerably improved the quality of the generated images to a level, where more details of the flower blossoms could be separated. Additional improvements to the model, implementing the cosine variance instead of the linear variance schedule, and increasing the data set size and the batch size for training could be used in the future to improve the results further.

Image augmentations tested in the scope of this project were not successful and resulted in very poor quality of the generated image. However, only one test was carried out without testing the hyperparameter values for the adjustments nor the adjustments separately. More detailed testing and including different adjustments such as rotation or shifting might show that some types of image adjustment improve the modeling process.

In general, the no-reference quality metrics calculated for the generated images agreed with the visual inspection. One exception of this was observed for the image augmentation testing, where CNN-IQA indicated the highest quality for the unconditional model across all tests done for the unconditional model. This was contrary to the visual evaluation of the images as well as the CNN-IQA score of the conditional model image augmentation test. There were additional examples of disagreement between the visual inspection and the quality metric scores, but these differences were on a smaller scale and might be resolved by generating more images during the inference.

5 Conclusion

This project explored the task of image generation based on the Oxford-Flowers data set and DDPM models. The models were implemented in PyTorch and a number of tests were carried out. Compared to the training batch size of 16, models' performance improved with the training batch size of 32. This improvement was confirmed by both visual evaluation of the generated images as well as three no-reference image quality metrics. Testing additional data augmentations on the training data set reduced the quality of the generated images. Additional testing and a more detailed analysis of specific image adjustments could be done in the future to understand why the applied augmentations lowered image quality and whether these adjustments in a different configuration or some other image augmentation methods could be beneficial for modeling. The two-layer U-Net architecture was extended to three-layer U-Net architecture with attention layers, which remarkably improved the quality of the generated images. Finally, FID and IS scores were calculated for sets of images generated with different guidance scales applied to the classifier-free guidance model with the three-layer U-Net model architecture. This was done to explore the trade-off between the adherence to the image labels and image diversity.

The analysis carried out in the scope of this project explored only a few specific parameters and model architectures. Nevertheless, the analysis of the generated images both visually and based on image quality metrics provided a good understanding of how DDPM models behave with respect to these specific changes to the model hyperparameters and architecture. This work can be seen as a good starting point for applying diffusion models in practice and the given project can be extended in the future for a number of additional analyses regarding model architectures, hyperparameters or new data sets.

6 Recommendations

In future winter schools, the focus of the tutorials could in addition to computer vision topics extend to other areas. Covering some of the state-of-the-art Natural Language Processing techniques could be beneficial to many. In addition, a discussion of how different types of data could be modelled with deep learning techniques could inspire students to look at their work from another angle. For example, modelling of genomic data or tabular data containing various types of features, such as environmental or clinical data, with deep learning methods could be discussed. The focus in these cases might not be on specific technologies or models but more on how to navigate the deep learning field from the perspective of input data and the task at hand. This more high-level discussion might be beneficial, as the field of deep learning is nowadays very broad. Therefore, it is highly important to have a structured way of approaching the task at hand, the input data and have an overview of the possibilities.

7 References

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022. arXiv: 2204.06125 [cs.CV].

[3] L. Yu, B. Shi, R. Pasunuru, B. Muller, O. Golovneva, T. Wang, A. Babu, B. Tang, B. Karrer, S. Sheynin, C. Ross, A. Polyak, R. Howes, V. Sharma, P. Xu, H. Tamoyan, O. Ashual, U. Singer, S.-W. Li, S. Zhang, R. James, G. Ghosh, Y. Taigman, M. Fazel-Zarandi, A. Celikyilmaz, L. Zettlemoyer, and A. Aghajanyan. *Scaling Autoregressive Multi-Modal Models: Pretraining and Instruction Tuning*. 2023. arXiv: 2309.02591 [cs.LG].

[4] S. Datta, A. Ku, D. Ramachandran, and P. Anderson. *Prompt Expansion for Adaptive Text-to-Image Generation*. 2023. arXiv: 2312.16720 [cs.CV].

[5] Y. Hao, Z. Chi, L. Dong, and F. Wei. *Optimizing Prompts for Text-to-Image Generation*. 2023. arXiv: 2212.09611 [cs.CL].

[6] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: 2112.10752 [cs.CV].

[7] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh. "Video generation models as world simulators". In: (2024). URL: https://openai.com/research/video-generation-models-as-world-simulators.

[8] H. M. Wang, C.-L. Tsai, and C.-T. Sun. "Automatic Generation of Game Content Customized by Players: Generate 3D Game Characters Based on Pictures". In: July 2023. ISBN: 978-1-83769-733-5. DOI: 10.5772/intechopen.1002024.

[9] J. Zhu, H. Yang, H. He, W. Wang, Z. Tuo, W.-H. Cheng, L. Gao, J. Song, and J. Fu. *MovieFactory: Automatic Movie Creation from Text using Large Generative Models for Language and Images*. 2023. arXiv: 2306.07257 [cs.CV].

[10] J. Ford, V. Jain, K. Wadhwani, and D. G. Gupta. "AI advertising: An overview and guidelines". In: *Journal of Business Research* 166 (2023), p. 114124. ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2023.114124. URL: https://www.sciencedirect.com/science/article/pii/S0148296323004836.

[11] H. Chung, E. S. Lee, and J. C. Ye. *MR Image Denoising and Super-Resolution Using Regularized Reverse Diffusion*. 2022. arXiv: 2203.12621 [eess.IV].

[12] M. Elasri, O. Elharrouss, S. Al-Maadeed, and H. Tairi. "Image Generation: A Review". en. In: *Neural Processing Letters* 54.5 (Oct. 2022), pp. 4609–4646. ISSN: 1573-773X. DOI: 10.1007/s11063-022-10777-x. URL: https://doi.org/10.1007/s11063-022-10777-x (visited on 03/09/2024).

[13] S. Frolov, T. Hinz, F. Raue, J. Hees, and A. Dengel. "Adversarial text-to-image synthesis: A review". In: *Neural Networks* 144 (2021), pp. 187–209. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2021.07.019. URL: https://www.sciencedirect.com/science/article/pii/S0893608021002823.

[14] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. arXiv: 1503.03585 [cs.LG].

[15] J. Ho, A. Jain, and P. Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG].

[16] P. Dhariwal and A. Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. arXiv: 2105.05233 [cs.LG].

[17] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". en. In: *Nature* 521.7553 (May 2015). Publisher: Nature Publishing Group, pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 03/09/2024).

[18] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. *Revisiting Unreasonable Effectiveness of Data in Deep Learning Era*. 2017. arXiv: 1707.02968 [cs.CV].

[19] J. Zhang, S. H. Yeung, Y. Shu, B. He, and W. Wang. *Efficient Memory Management for GPU-based Deep Learning Systems*. 2019. arXiv: 1903.06631 [cs.DC].

[20] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen. *Image Data Augmentation for Deep Learning: A Survey*. 2023. arXiv: 2204.08610 [cs.CV].

[21] G. E. Hinton, S. Osindero, and Y.-W. Teh. "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Computation* 18.7 (July 2006), pp. 1527–1554. ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527. eprint: https://direct.mit.edu/neco/article-pdf/18/7/1527/816558/neco.2006.18.7.1527.pdf. URL: https://doi.org/10.1162/neco.2006.18.7.1527.

[22] I. H. Sarker. "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions". en. In: *SN Computer Science* 2.6 (Aug. 2021), p. 420. ISSN: 2661-8907. DOI: 10.1007/s42979-021-00815-1. URL: https://doi.org/10.1007/s42979-021-00815-1 (visited on 03/09/2024).

[23] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. en. Ed. by G. James, D. Witten, T. Hastie, and R. Tibshirani. Springer Texts in Statistics. New York, NY: Springer US, 2021. ISBN: 978-1-07-161418-1. DOI: 10.1007/978-1-0716-1418-1_1. URL: https://doi.org/10.1007/978-1-0716-1418-1_1 (visited on 03/09/2024).

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[25] A. F. Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2019. arXiv: 1803.08375 [cs.NE].

[26] A. Zafar, M. Aamir, N. Mohd Nawi, A. Arshad, S. Riaz, A. Alruban, A. K. Dutta, and S. Almotairi. "A Comparison of Pooling Methods for Convolutional Neural Networks". In: *Applied Sciences* 12.17 (2022). ISSN: 2076-3417. DOI: 10.3390/app12178643. URL: https://www.mdpi.com/2076-3417/12/17/8643.

[27] L. Bottou and O. Bousquet. "The Tradeoffs of Large Scale Learning". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc., 2007. URL: https://proceedings.neurips.cc/paper_files/paper/2007/file/0d3180d672e08b4c5312dcdafdf6ef36-Paper.pdf.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". en. In: *Nature* 323.6088 (Oct. 1986). Publisher: Nature Publishing Group, pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: https://www.nature.com/articles/323533a0 (visited on 03/09/2024).

[29] Y. Wu, L. Liu, J. Bae, K.-H. Chow, A. Iyengar, C. Pu, W. Wei, L. Yu, and Q. Zhang. *Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks*. 2019. arXiv: 1908.06477 [cs.LG].

[30] J. Kukačka, V. Golkov, and D. Cremers. *Regularization for Deep Learning: A Taxonomy*. 2017. arXiv: 1710.10686 [cs.LG].

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[32] Y. Song and S. Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. 2020. arXiv: 1907.05600 [cs.LG].

[33] D. P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML].

[34] J. Ho and T. Salimans. *Classifier-Free Diffusion Guidance*. 2022. arXiv: 2207.12598 [cs.LG].

[35] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. *Cascaded Diffusion Models for High Fidelity Image Generation*. 2021. arXiv: 2106.15282 [cs.CV].

[36] A. Shih, S. Belkhale, S. Ermon, D. Sadigh, and N. Anari. *Parallel Sampling of Diffusion Models*. 2023. arXiv: 2305.16317 [cs.LG].

[37] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. "Video diffusion models". In: *arXiv:2204.03458* (2022).

[38] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang. *SmartBrush: Text and Shape Guided Object Inpainting with Diffusion Model*. 2022. arXiv: 2212.05034 [cs.CV].

[39] X. Li, Y. Ren, X. Jin, C. Lan, X. Wang, W. Zeng, X. Wang, and Z. Chen. *Diffusion Models for Image Restoration and Enhancement – A Comprehensive Survey*. 2023. arXiv: 2308.09388 [cs.CV].

[40] Y. Huang, J. Huang, Y. Liu, M. Yan, J. Lv, J. Liu, W. Xiong, H. Zhang, S. Chen, and L. Cao. *Diffusion Model-Based Image Editing: A Survey*. 2024. arXiv: 2402.17525 [cs.CV].

[41] J. M. L. Alcaraz and N. Strodthoff. *Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models*. 2023. arXiv: 2208.09399 [cs.LG].

[42] N. Anand and T. Achim. *Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models*. 2022. arXiv: 2205.15019 [q-bio.QM].

[43] C. Cao, Z.-X. Cui, Y. Wang, S. Liu, T. Chen, H. Zheng, D. Liang, and Y. Zhu. "High-Frequency Space Diffusion Model for Accelerated MRI". In: *IEEE Transactions on Medical Imaging* (2024). ISSN: 1558-254X. DOI: 10.1109/tmi.2024.3351702. URL: http://dx.doi.org/10.1109/TMI.2024.3351702.

[44] T. Chen. *On the Importance of Noise Scheduling for Diffusion Models*. 2023. arXiv: 2301.10972 [cs.CV].

[45] D. S. Lemons. *An Introduction to Stochastic Processes in Physics*. en. Johns Hopkins University Press, 2002. ISBN: 978-0-8018-6867-2 978-0-8018-7638-7 978-0-8018-6866-5. DOI: 10.56021/9780801868665. URL: https://www.press.jhu.edu/books/title/1132/introduction-stochastic-processes-physics (visited on 03/09/2024).

[46] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[47] J. Long, E. Shelhamer, and T. Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV].

[48] Y. Peng, M. Sonka, and D. Z. Chen. *U-Net v2: Rethinking the Skip Connections of U-Net for Medical Image Segmentation*. 2023. arXiv: 2311.17791 [eess.IV].

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[50] A. Nichol and P. Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. 2021. arXiv: 2102.09672 [cs.LG].

[51] *carmenoroperv/DDPM_oxford_flowers: DDPM and DDPM with classifier-free guidance implemented for the NLDL 2024 winter school project*. URL: https://github.com/carmenoroperv/DDPM_oxford_flowers (visited on 03/10/2024).

[52] Cloneofsimo. *github/Cloneofsimo/Mindiffusion: Self-contained, minimalistic implementation of diffusion models with Pytorch*. 2022. URL: https://github.com/cloneofsimo/minDiffusion.

[53] FilippoMB. *github/FilippoMB /Diffusion$_m$odels$_t$utorial*. 2022. URL: https://github.com/FilippoMB/Diffusion_models_tutorial.

[54] TeaPearce. *conditional$_d$iffusion$_m$nist*. 2022. URL: https://github.com/TeaPearce/Conditional_Diffusion_MNIST/tree/main.

[55] L. Kang, P. Ye, Y. Li, and D. Doermann. "Convolutional Neural Networks for No-Reference Image Quality Assessment". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1733–1740. DOI: 10.1109/CVPR.2014.224.

[56] A. Mittal, R. Soundararajan, and A. C. Bovik. "Making a "Completely Blind" Image Quality Analyzer". In: *IEEE Signal Processing Letters* 20.3 (2013), pp. 209–212. DOI: 10.1109/LSP.2012.2227726.

[57] A. Mittal, A. K. Moorthy, and A. C. Bovik. "No-Reference Image Quality Assessment in the Spatial Domain". In: *IEEE Transactions on Image Processing* 21.12 (2012), pp. 4695–4708. DOI: 10.1109/TIP.2012.2214050.

[58] *huggan/flowers-102-categories · Datasets at Hugging Face*. Aug. 2022. URL: https://huggingface.co/datasets/huggan/flowers-102-categories (visited on 03/10/2024).

[59] *coroperv/oxford-flowers · Datasets at Hugging Face*. URL: https://huggingface.co/datasets/coroperv/oxford-flowers (visited on 03/10/2024).

[60] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG].

[61] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. "Improved Techniques for Training GANs". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf.

[62] S. Jayasumana, S. Ramalingam, A. Veit, D. Glasner, A. Chakrabarti, and S. Kumar. *Rethinking FID: Towards a Better Evaluation Metric for Image Generation*. 2024. arXiv: 2401.09603 [cs.CV].

[63] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, and T. Wolf. *Diffusers: State-of-the-art diffusion models*. https://github.com/huggingface/diffusers. 2022.