Efficient CIFAR-100 Modelling

Alexander Read (jjxc38)

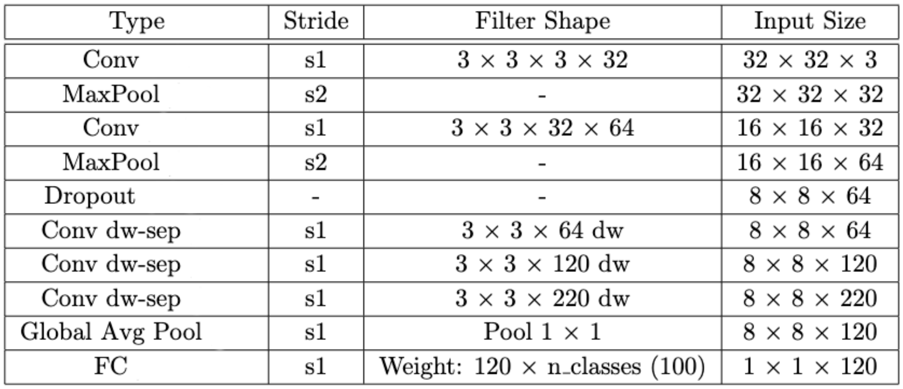
Abstract

This paper proposes a compact architecture for CIFAR-100 image classification, employing fewer than 100,000 parameters and combines standard and depth-wise separable convolutional layers. Additionally, we introduce a low-parameter diffusion model for image generation on the CIFAR-100 dataset featuring a UNet-based architecture that uses modified ConvNext blocks, classifier-free guidance, and an estimated moving average.

Part 1: Classification - Baby Mobile Net

1 Methodology

Figure 1 presents our deep classification network, incorporating both standard convolutional layers and Depthwise Separable Layers, drawing inspiration from the original mobile net paper [6].



Type	Stride	Filter Shape	Input Size
Conv	s1	3 × 3 × 3 × 32	32 × 32 × 3
MaxPool	s2	-	32 × 32 × 32
Conv	s1	3 × 3 × 32 × 64	16 × 16 × 32
MaxPool	s2	-	16 × 16 × 64
Dropout	-	-	8 × 8 × 64
Conv dw-sep	s1	3 × 3 × 64 dw	8 × 8 × 64
Conv dw-sep	s1	3 × 3 × 120 dw	8 × 8 × 120
Conv dw-sep	s1	3 × 3 × 220 dw	8 × 8 × 220
Global Avg Pool	s1	Pool 1 × 1	8 × 8 × 120
FC	s1	Weight: 120 × n_classes (100)	1 × 1 × 120

Figure 1: Classifier Architechure

A Depthwise convolution with one filter per input channel can defined as:

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m}$$

Where $\hat{K}$ is the depthwise convolutional kernel of size $D_K \times D_K \times M$ where the mth filter in $\hat{K}$ is applied to the mth channel in F to produce the mth channel of the filtered output feature map $\hat{G}$.

A depthwise separable convolution uses up to 9 times fewer parameters than a standard convolution in our model as they apply a depthwise convolution followed by a pointwise convolution.

$$\begin{aligned}\text{Difference in parameter cost(defined in [6]):} &= \frac{\text{Depthwise Separable Convolution Cost}}{\text{Standard Convolution Cost}}\\ &= \frac{DK \cdot DK \cdot M \cdot DF \cdot DF + M \cdot N \cdot DF \cdot DF}{DK \cdot DK \cdot M \cdot N \cdot DF \cdot DF}\\ &= \frac{1}{N} + \frac{1}{DK^2}\end{aligned}$$

Therefore, the cost is approximately 11.9% of the cost of a standard convolution (best case), which can be calculated as $\frac{1}{120} + \frac{1}{9}$.

Where: M = Number of input channels, N = Number of output channels, DK = Kernel size (assuming square, $DK \times DK$), DF = Feature map size (assuming square, $DF \times DF$).

Despite the low parameter count of depthwise separable convolutions, the architecture (seen in Figure 1) starts with two standard convolutional layers, as these layers are better at the extraction of basic shapes and textures [6], whilst depthwise separable layers are used in the deeper layers. The model uses a ReLU activation function [1], $\text{ReLU}(x) = \max(0, x)$, and a dropout layer is used to reduce overfitting.

Further optimisations included increased Batch size B to 512 for less noisy gradient updates and faster convergence, enabling the small architecture to stabilize more effectively [13].

Data augmentation applied to the training data included random crop, horizontal flip, rotation, colour jitter, gaussian blur, normalization, and random erasing. These methods reduced overfitting and variance.

A learning rate scheduler was used for faster convergence. We found the step LR [8] scheduler

$$\eta(t) = \eta_0 \cdot \alpha^t$$

Where: - $\eta(t)$ represents the learning rate at time t, & η_0 is the initial learning rate;
to be effective at increasing test accuracy whilst reducing variance.

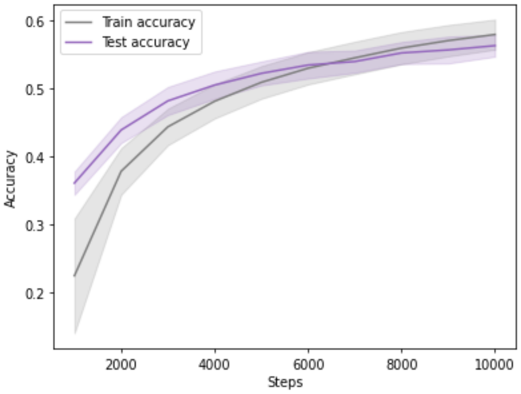
Optimisations improved the model's test accuracy by 13%

2 Results

The network has 97,584 parameters.

It attains 57.9% training accuracy and also 56.3% testing accuracy at 10,000 optimisation steps (train loss: 1.507, train acc: 0.579±0.022, test acc: 0.563±0.016). The model has a variance of under 2%.

The training graph:



3 Limitations

The results are promising; however, the parameter constraints limit the model. This architecture can scale with additional parameters/layers, as seen in iterations of MobileNet [6] like MobileNet v3 [5], which employ millions more parameters with use of depth-wise separable layers and a similar architecture structure. Such models have achieved over 70% accuracy on CIFAR-100 [7].

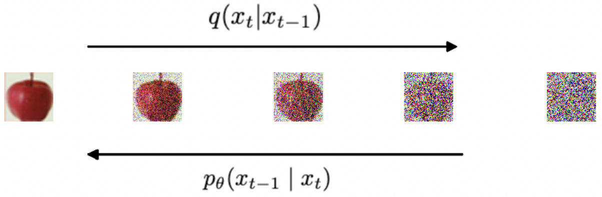
Part 2: Generative model - How Do You Like Them Apples?

4 Methodology

We develop a denoising diffusion model (DDPM) [4] Where a forward process q adds Gaussian noise to an image x over 1000 steps (T = 1000) using a linear variance schedule, $q(x_t|x_{t-1})$.

A neural network is trained to learn the reverse denoising diffusion process, p_θ by learning the difference in noise between an image at two steps t and $t-1$.

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$



where the mean μ_θ depends on the noise level t and the variance Σ_θ is fixed.

We use the Unet architecture [12], which works by downsampling and upsampling noisy images to learn their key features and guess the amount of noise applied to the original image. This process is shown in Figure 2 . The objective function used to train the model can be defined as $\mathcal{L} = \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$. We use the Mean Squared Error (MSE) loss function $MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$.

The architecture receives the image's noise level (time step t) through a sinusoidal position embedding $s(x_t)$, with the noised image x_t. We add conditional DDPM by embedding the image's class label and adding it to the sinusoidal position embedding.

$$\text{input} = s(x_t) + emb(\text{class\_label}), x_t$$

In our modified Unet architecture, we implement ConvNeXt blocks [9] featuring more depthwise separable convolutions [6] (see the previous classifier section). This replaces the standard ResNet blocks [12] used in the original Unet design. ConvNeXt blocks use depthwise separable convolutions and layer normalization and have a lower parameter count than ResNet blocks. We further streamline the Unet model by substituting more standard convolutions with depthwise separable convolutions in the ConvNeXt blocks and omitting inner layers of the Unet architecture that are typically bypassed via skip connections.

We implement classifier-free guidance by presenting the model with an image without a label 10% of the time whilst training. This allows the model to learn the general patterns of the images (unconditional training). We implement exponential moving average (EMA) to smooth model updates and reduce the effects of outliers. The EMA takes a copy of the model weights (w) at each cycle of 2500 steps and saves new model weights defined as:

$w = \beta \cdot w_{\text{old}} + (1-\beta) \cdot w_{\text{new}}$, where $\beta = 0.998$

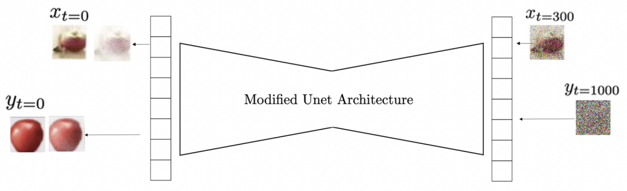
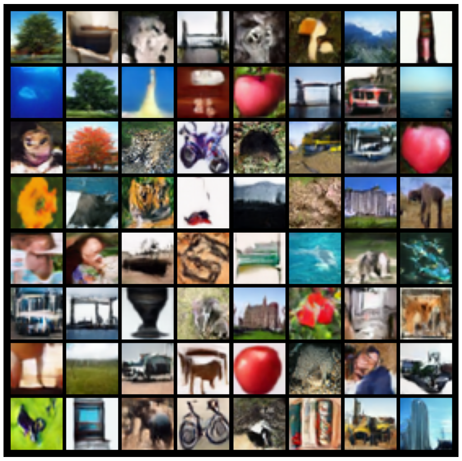


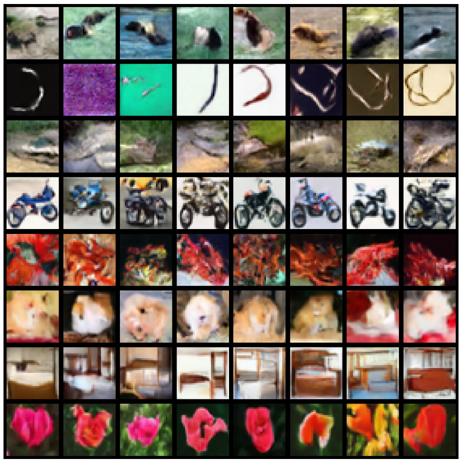
Figure 2: How diffusion process works with Unet model

5 Results

The network has 960,223 parameters and achieves an FID of 25.79 against the CIFAR-100 test dataset. It was trained for 50,000 optimisation steps.

The results are impressive, particularly with the parameter constraint; the sample set is diverse and models most of the dataset; the sampled images do not appear to suffer from mode collapse. A random batch of non-cherry-picked samples looks like this (left), and interpolants between points in the latent space look like this (right), due to the complexity of diffusion model interpolations, linear interpolations between the same class were employed for this example.





And here are some cherry-picked samples that show the best outputs the model has generated:

6 Limitations

The model's limitations arise from constraints on parameters and training duration. Including more ConvNeXT blocks and expanding the Unet structure could enable the model to produce more detailed images and recognise more intricate features. Adopting an alternate variance scheduler like a cosine scheduler, which has demonstrated superior efficacy [3], could be used for further improvements. Interpolating between images in diffusion models can be challenging because the latent space in these models may not be as smoothly structured or 'flat' as in GANs, because of this a further improvement would be to implement more advanced interpolation techniques [14]. Techniques in leading diffusion models like Stable Diffusion [11] and DALLE-2 [10] involve estimating the variance and mean instead of maintaining a fixed variance Σ_θ as seen in this model. This method is shown to produce more accurate results [2].

References

[1] Abien Fred Agarap. "Deep learning using rectified linear units (relu)". In: *arXiv preprint arXiv:1803.08375* (2018).

[2] Fan Bao et al. "Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models". In: *arXiv preprint arXiv:2201.06503* (2022).

[3] Hugging Face. *Annotated Diffusion*. `https://huggingface.co/blog/annotated-diffusion`. Accessed: January 31, 2024. 2023.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[5] Andrew Howard et al. "Searching for mobilenetv3". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1314–1324.

[6] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[7] *Image Classification on CIFAR-100 - Papers with Code*. Accessed on January 30, 2024. URL: `https://paperswithcode.com/sota/image-classification-on-cifar-100`.

[8] Chiheon Kim et al. "Automated learning rate scheduler for large-batch training". In: *arXiv preprint arXiv:2107.05855* (2021).

[9] Zhuang Liu et al. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.

[10] Aditya Ramesh et al. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022. arXiv: `2204.06125 [cs.CV]`.

[11] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: `2112.10752 [cs.CV]`.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[13] Leslie N Smith and Nicholay Topin. "Super-convergence: Very fast training of neural networks using large learning rates". In: *Artificial intelligence and machine learning for multi-domain operations applications*. Vol. 11006. SPIE. 2019, pp. 369–386.

[14] Clinton J Wang and Polina Golland. "Interpolating between images with diffusion models". In: *arXiv preprint arXiv:2307.12560* (2023).