Convolutional Variational Autoencoders for Generation of Human Faces

A. J. Manlove

All images, figures and diagrams in this report are of my own creation. This project and the relevant data are hosted on GitHub: `github.com/alexjmanlove/convolutional-variational-autoencoders`

Project Overview

This project explores the application of convolutional neural networks (CNNs) in the domain of human face images. In the first chapter, CNNs are used to perform supervised prediction of the gender, ethnicity and age of a person in an image. The second chapter evaluates the application of variational autoencoder (VAE) models with convolutional encoder and deconvolutional decoder networks to perform unsupervised tasks, such as restoration of corrupted images and interpolation. We investigate how the VAE hyperparameters of latent dimension size and regularisation scaling coefficient affect the reconstructions.

Contents

Chapter 1

Classification and Regression with Convolutional Neural Networks

1.1 Brief Recap of Convolutional Neural Networks

Convolutional neural networks (CNNs) are artificial neural networks which use convolutions to analyse the spatial relationships between pixels in an image, allowing the network to extract spatial structures to be used as features for tasks downstream [1: Lecun et al., 1998]. Convolution involves sliding a filter along the input image and computing the dot products between the filter and the local regions of the image. This repeated operation outputs a feature map that encodes the presence or absence of spatial patterns. Fig 1.1, below, illustrates stages of simple CNN usage to perform binary image classification.

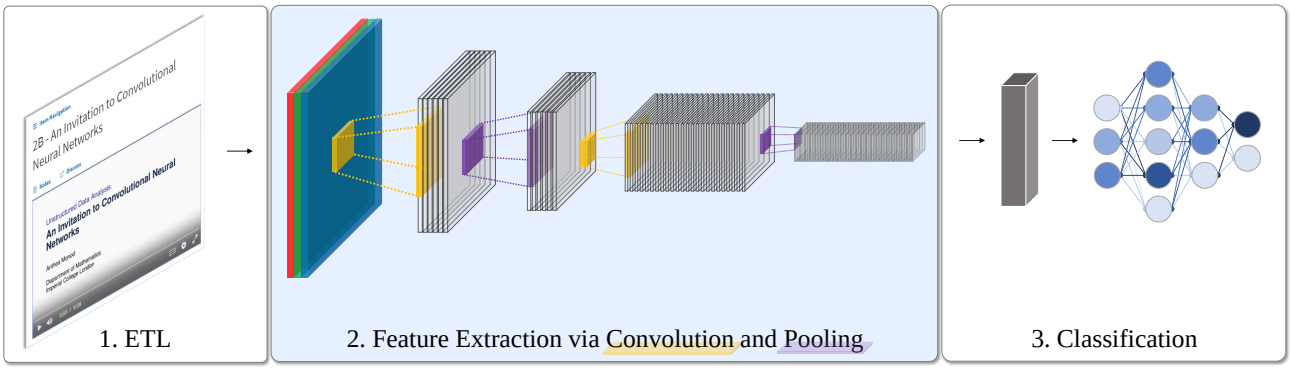


Figure 1.1: Visualising the application of a simple CNN for binary image classification.

Input images are initially transformed into rank 3 tensors, with height and width corresponding to pixel count, and depth equal to the number of colour channels. CNNs typically contain multiple pairs of convolution and pooling layers. Pooling layers are used to downsample the feature maps by taking the max, min, or average value over a local region. This reduces the height and width of the feature maps, making them more computationally efficient to process. This approach improves the robustness of the model to small translations or perturbations in the input image. Once the features have been extracted, they are typically flattened and fed into a fully-connected network for the prediction task.

1.2 Data Introduction and Exploratory Analysis

The dataset of interest contains details of $27,305$ greyscale images of dimensions 48×48 pixels and is derived from the UTKFace dataset, which contains the original larger-size full-colour images: `kaggle.com/datasets/nipunarora8/age-gender-and-ethnicity-face-data-csv`. The images in the data are labelled with two nominal categorical variables and one integer variable:

1. GENDER of the individual. This is coded as a binary integer with 0 for male, and 1 for female.
2. ETHNICITY of the individual, belonging exclusively to one of five classes:
   { 0: White, 1: Black, 2: East Asian, 3: South Asian, 4: Latino }.
3. AGE of the individual, where age $\in [0, 116] \subset \mathbb{Z}$.

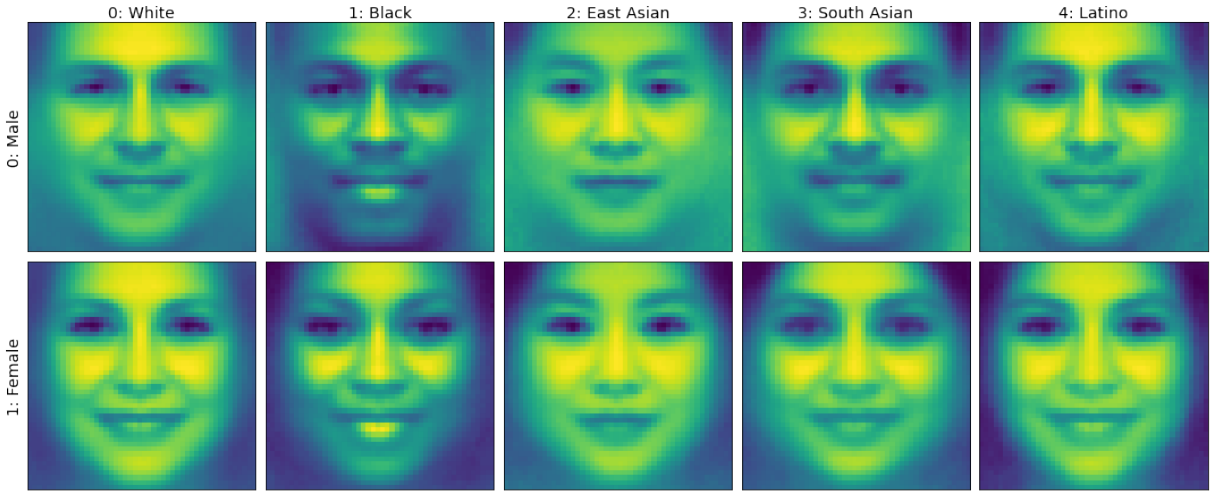


Figure 1.2: Sample means corresponding to one segment per Gender-Ethnicity class. The intensity value at each pixel is determined by the mean intensity value of all pixels with shared coordinate in the segment.

Fig. 1.2 above, containing sample means for each Gender-Ethnicity pairing, lends us intuition about the perceptible differences in features across classes in aggregate. Likewise, Fig. 1.3, to the right, provides another visualisation of sample means. In this figure, the data are again partitioned by gender, but with further segmentation into age buckets. Qualitatively, this figure reveals a potential interdependence between our variables. In particular, class separability along gender appears qualitatively poorer for observations with age < 12. By contrast, the separability between observations associated to older subjects appears discernibly more pronounced.

Moreover, a brief exploratory analysis of our data reveals a potentially problematic class imbalance along the variable ETHNICITY. This can be seen in the centre bar plot below in Fig 1.4. Specifically, the data showed that a disproportionate number of observations belonged to ethnic group 0, while other ethnic groups were underrepresented. Likewise there is a lack of data in the upper age ranges. These imbalances could pose challenges when developing the models, as biased training data can lead to biased predictions.

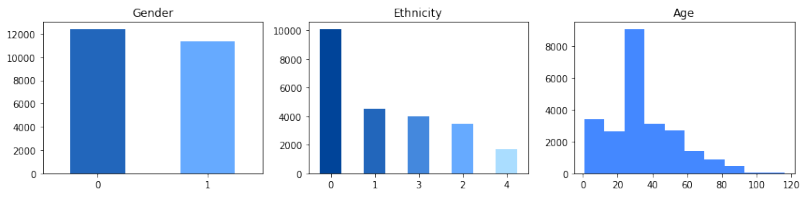


Figure 1.4: Plot of distributions reveals class imbalance along ETHNICITY and over-representation of AGEs in the range 25-35.

Ultimately these insights are noteworthy because our models will rely upon discerning the small differences highlighted in figures 1.2 and 1.3 in order to make predictions. Where class separability appears poor, or the data is afflicted with severe imbalances, models may struggle.

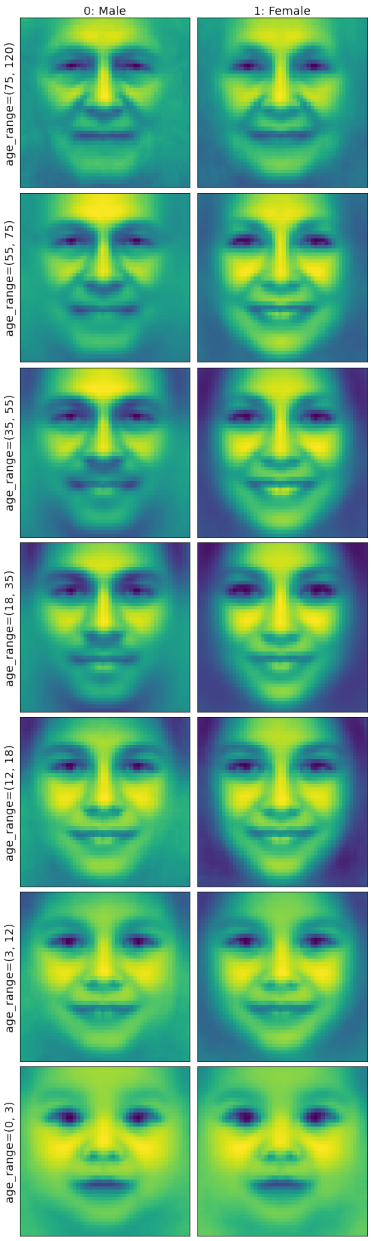


Figure 1.3: Grid of sample means over all ethnicities, segmented by gender and age.

1.3 CNN Design and Training

We will compare the training and test performance of three convolutional networks. The first two of these will be trained to perform classification of gender and of ethnicity. The third will be trained to perform regression to predict the subject's age.

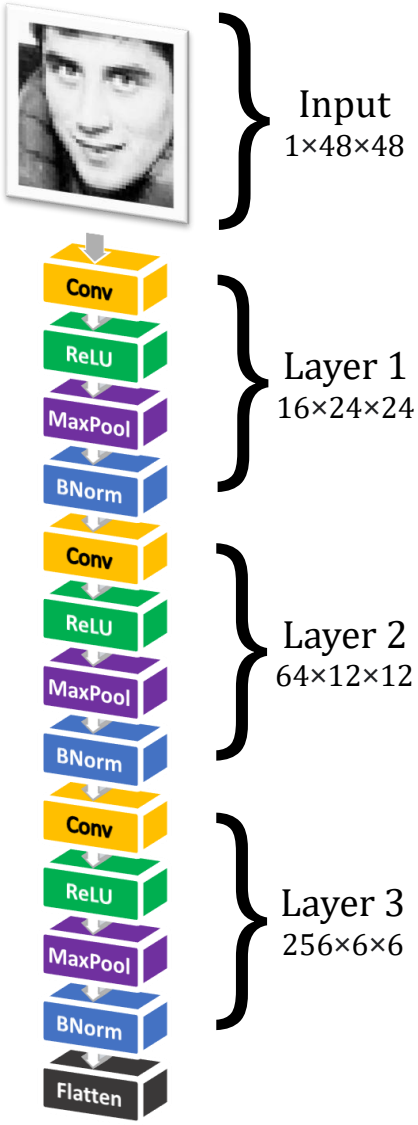


Figure 1.5: Feature extraction template used by all models.

1.3.1 Feature Extraction

We initialise each of the three CNNs with identical feature extraction architectures. The difference in task objective is expressed purely in the specification of the assigned loss functions and the prediction networks downstream.

The networks are initialised with three pairs of convolution and pooling layers. ReLU activation is applied after each convolution. Batch normalisation is applied after pooling to assist training [2, Ioffe & Szegedy, 2015]. Fig. 1.5 to the left provides an illustration of these feature mapping layers. The flattened outputs are fed into the prediction networks.

The plot below in Fig. 1.6 shows example outputs at each layer. In this figure we can see how specific spatial structures, such as the edge of the face or the eyes, are captured by the convolutional filters at the first layer. After the third layer, the outputs correspond to more abstract features that are less directly tied to specific spatial structures but which capture higher-level information about the input image.

1.3.2 Classification, Regression and Loss

The two classification networks apply softmax activation at the final layer to output a vector of predicted class membership probabilities for each input image, with dimension equal to the number of available classes. Cross entropy loss is used for the training of these models. Similarly, the regression network outputs a single scalar value per observation, predicting the age of the subject. This network will be optimised to reduce the mean squared error or L2 loss over these age predictions. All three prediction networks downstream make use of LeakyReLU activation.

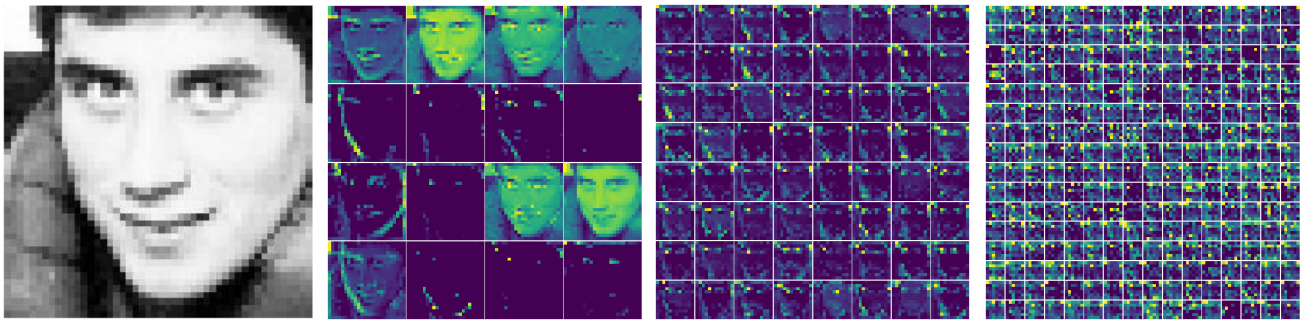


Figure 1.6: Output at each feature mapping layer.

To perform classification, a convolutional neural network $\mathcal{C} : X \mapsto \hat{P}$, maps the rank 3 input tensor X with dimensions $N \times H \times W$ to the rank 2 output tensor $\hat{P}$ with dimensions $N \times K$, where $H \times W$ are the dimensions of the input images and K is the number of available classes. The $(i, j)^{th}$ element of $\hat{P}$, contains the predicted membership probability $\hat{p}_{ij}$ of observation $i \in \{1, ..., N\}$ to class $j \in \{1, ..., K\}$, such that $\sum_{j=1}^{K} \hat{p}_{ij} = 1$. Similarly, the output of the regression model $\mathcal{R} : X \mapsto \hat{A}$ is a rank 1 tensor $\hat{A}$ of length N predicting the ground truth ages $A = [a_1, ..., a_N]^T \in \mathbb{Z}^N$. To train the models $\mathcal{C}$ and $\mathcal{R}$, we can minimise the cross entropy loss L_{CE} and mean squared error L_{MSE} of predictions:

$$L_{\text{CE}}(\hat{P}; Y) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} y_{ij} \log(\hat{p}_{ij}), \qquad L_{\text{MSE}}(\hat{A}; A) = \frac{1}{N} \sum_{i=1}^{N} \|a_i - \hat{a}_i\|^2, \tag{1.1}$$

where y_{ij} of matrix Y is the boolean ground truth for membership of the i^{th} sample to j^{th} class.

1.3.3 Training, Learning Rate Schedule and Validation Loss

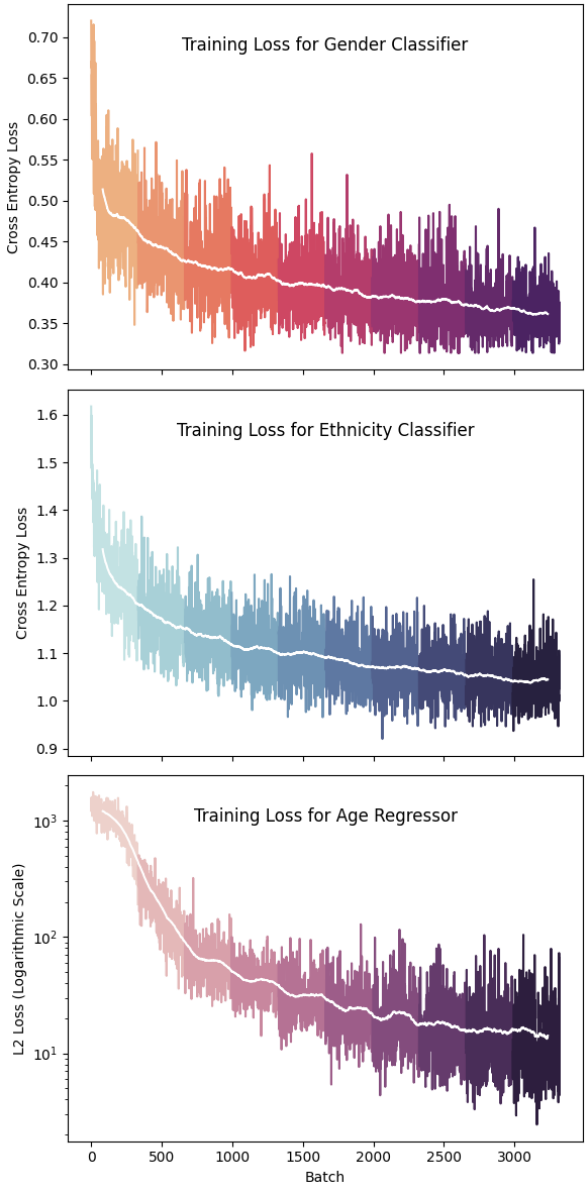


Figure 1.7: Training losses using learning rate scheduler over 10 epochs. Running means have been overlaid in white to highlight trend.

We perform a random split of our data into training, validation and test sets, in the ratio 7:2:1. The three models are allowed to train for 10 epochs. A learning rate scheduler is used to assist model training. Learning rate is a hyperparameter that determines the step-size taken by the optimiser during gradient descent. A scheduler is a mechanism that adjusts this learning rate throughout the training cycle, which is useful because the optimal learning rate may change as the model trains. Tuning was performed with several candidate functions to determine the optimal scheduler. Here, we determined that a simple schedule function $r : \mathbb{N} \to \mathbb{R}$, $r(E) = 2000^{-1} \times (1 + E)^{-1}$ provided a reasonable learning rate at each epoch $E \in \{0, 1, ..., 9\}$.

Plots of the training loss for each model, using this scheduler, are shown in Fig. 1.7 to the left, coloured by epoch. Plots of the classifier accuracy and regressor MSE are shown in Fig 1.8 below.

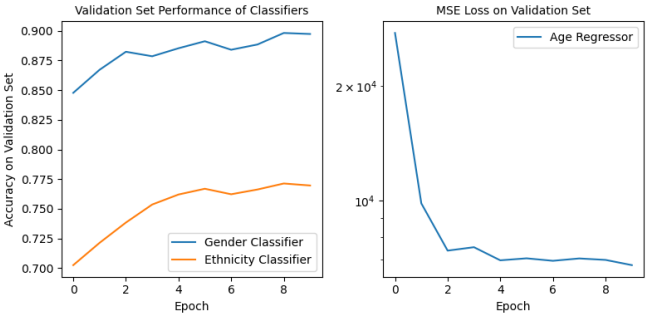


Figure 1.8: Model performances on validation set.

These plots suggest the models have successfully learnt from the training data to reduce bias without overfitting; the performance on the validation set continues to improve while training loss decreases.

1.4 Final Evaluation of Supervised Models on Test Set

Figure 1.9: Confusion matrices for the GENDER and ETHNICITY classifiers on test data.

The two classifiers have successfully achieved near 90% and 78% accuracy on the test set. However, as suspected previously, the data imbalance has resulted in poorer performance for the ethnicity classifier when classifying observations in the minority classes. This is highlighted in the confusion matrices above in Fig 1.9, as well as in the derived metrics shown below in Fig 1.10. A potential solution to this issue would be to retrain the model on an augmented dataset. We could augment the data, for example using flipping or shearing, to generate additional samples of observations in the minority classes.

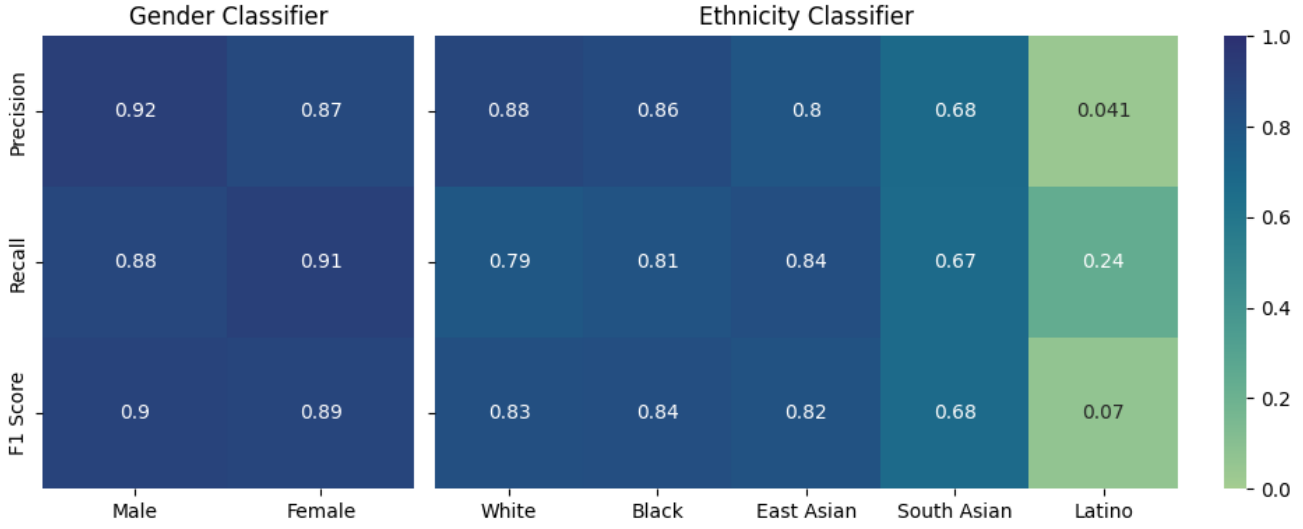
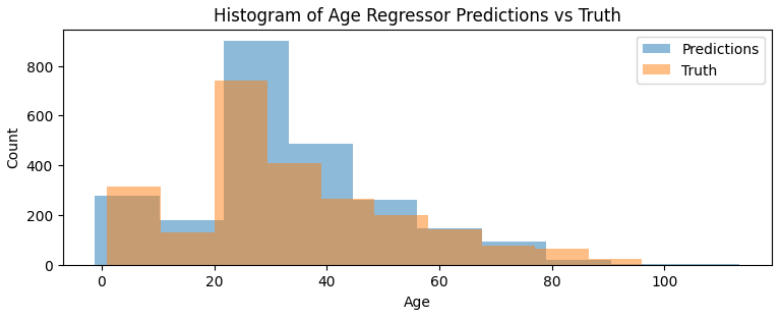


Figure 1.10: Precision, Recall and F1 Score for GENDER and ETHNICITY classifiers.

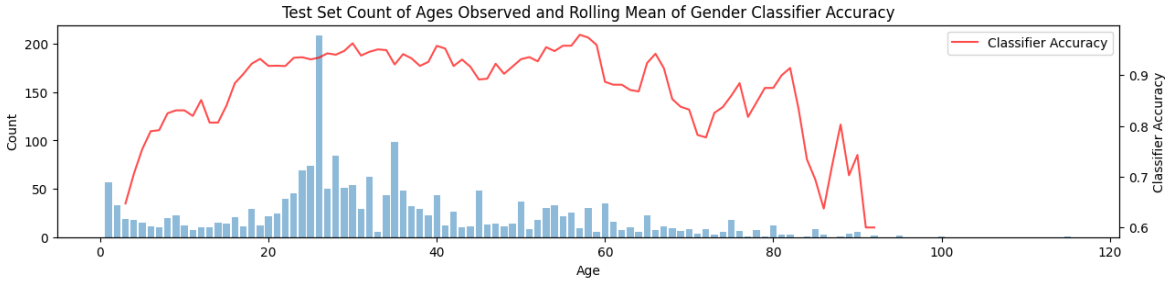
Furthermore, the age regression predictions appear to adhere to the true distribution of the data, which is highlighted on the following page in the histograms in Fig 1.11. However, the predictions appear slightly positively skewed, with very few predictions towards the upper tail of age ranges and a disproportionately high density of predictions around the mean. This is possibly a reflection of the bias issues highlighted in the exploratory analysis. A **Mean Absolute Error** of **6.0315** and a **Root Mean Squared Error** of **8.5451** were recorded on the aggregate test set.

Figure 1.11: Histograms of predicted vs observed age reveal clear overlap between model outputs and the ground truth distribution of the data.



To investigate this further we consider the impact of a subject's age on test accuracy of the gender classifier model. As demonstrated in Fig. 1.12, below, the accuracy varies by age. It can be observed that the model accuracy is relatively poorer at both extremes of the age range, whereas peak performance is observed around the median, in the age range of 20 to 60. Previous exploratory visualisation, shown in Section 1.2, revealed poorer class separability along gender for young individuals (age < 12), which may explain the classifier's weak performance for observations with AGE in the first quintile.

Figure 1.12: Gender classifier accuracy is strong around the median age and weakest at the extremes.



The performance of the gender classifier increases with the age of subjects, reaching a plateau in the range 20 to 40. However, at 60 years and beyond, the performance of the classifier declines, reaching minimum accuracy at the uppermost age ranges. This is likely a further consequence of the data imbalance issues highlighted previously. The bar plot in Fig. 1.12, showing the density of observations by age, emphasises the critical lack of data for observations with age 80 and up. This was a clear limitation of the dataset and hinders the ability of the classifier to properly learn the features that distinguish older subjects. Sample predictions using all models are shown below, in Fig. 1.13.

Figure 1.13: Ground truth versus model predictions on 12 random samples from the test set.

Chapter 2

Image Reconstruction and Generation with Variational Autoencoders

2.1 Brief Recap of Autoencoders

Autoencoders are unsupervised models that use an encoder-decoder architecture to reconstruct unlabelled input data. We formulate the autoencoder as a pair of networks with parameters θ and ϕ. The encoder E_ϕ maps a high dimensional input x to a low dimensional latent representation z. Conversely, the decoder network D_θ maps the compact latent code back to the original input space. An example illustration is shown below in Fig. 2.1. The objective is to learn the parameters θ and ϕ of the encoder and decoder networks, such that the resulting reconstruction $\hat{x} = D_\theta(E_\phi(x))$ is as close as possible to the original data x. In the case of image data, common loss functions include the MSE or cross entropy loss, although the peak signal-to-noise ratio and structural similarity index are also widely used.

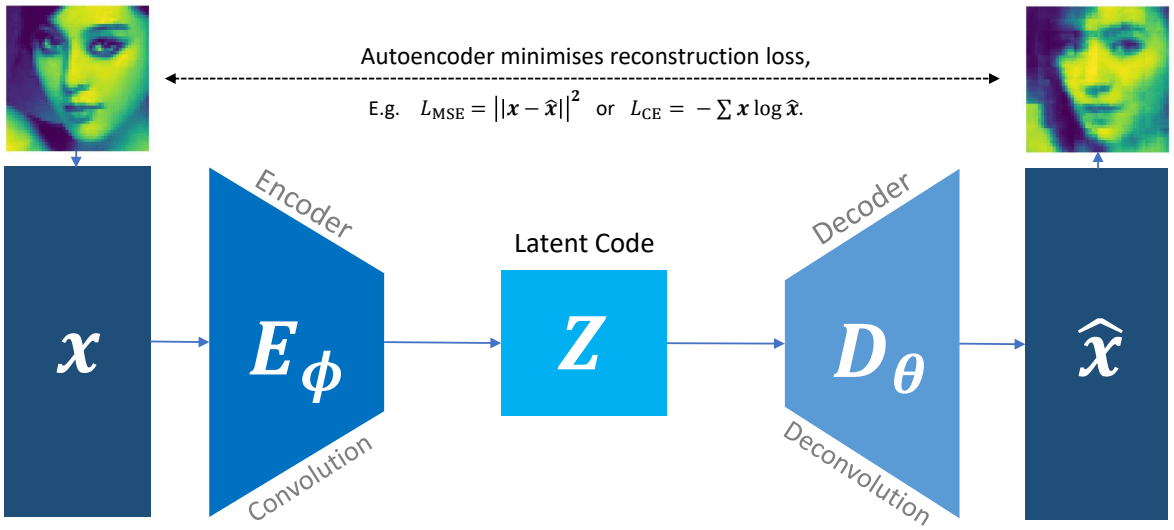


Figure 2.1: In this example autoencoder, E_ϕ is a CNN which compresses the image x down to a compact latent code z. The decoder D_θ performs deconvolutions to reconstruct the image from the latent code.

The decoder D_ϕ uses transposed convolution, or deconvolution, in order to upsample the latent vectors to high resolution images. Deconvolution involves applying the tensor transpose of the convolution kernel to the input tensor, with a stride that determines the size of the output tensor. As the kernel is applied to the input tensor, overlapping operations increase the height and width of the tensor while reducing the number of channels. This results in an upsampling of the tensor and an increase in resolution.

2.1.1 Latent Dimensions

The number of permitted latent dimensions is a hyperparameter that determines the size of the latent space. This hyperparameter presents a bias-variance tradeoff. If the latent dimensions are too few, the autoencoder will lack the sophistication to capture the full complexity of the input data, which can lead to poor quality reconstructions. On the other hand, if the latent dimensions are too numerous, the model may overfit to the training set and generalise poorly on unseen data.

2.2 Variational Autoencoders

The key difference between VAEs and traditional autoencoders is that the latent representation learned by a VAE is not a fixed, deterministic encoding of the input data. Rather, VAEs build on the approach of traditional autoencoders by incorporating principles of probabilistic generative modelling and Bayesian variational inference to learn a distribution over the possible latent codes [3: Kingma et al., 2013]. This approach involves specifying a prior $q(z)$ over the latent variables, which guides the latent representations learned by the encoder. We seek to minimise the loss function $\mathcal{L}$, which balances two objectives:

$$\underset{\theta,\phi}{\operatorname{argmin}}\, \mathcal{L}(x, \hat{x}; \theta, \phi), \qquad \mathcal{L}(x, \hat{x}; \theta, \phi) = \beta\ D_{\mathrm{KL}}[q_\phi(z|x)|q(z)] - \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]. \tag{2.1}$$

The term $D_{\mathrm{KL}}[q_\phi(z|x)|q(z)]$ is the Kullback-Leibler (KL) divergence between the posterior encoder distribution $q_\phi(z|x)$ and the prior $q(z)$. In effect, this regularisation term constrains the posterior latent variable distribution, encouraging it to adhere to our specified prior. Here we suppose a standard Gaussian prior with diagonal covariance matrix. To control the relative contribution of this term we introduce the hyperparameter coefficient β, which scales the strength of the regularisation. This can also help to prevent posterior collapse, which is a phenomenon where the approximate posterior becomes too narrow and fails to capture the diversity of the data [4: Chou, 2019].

The reconstruction loss term $\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$ is the expected reconstruction log-likelihood, which is a measure of the quality of the reconstructions produced by the decoder network p_θ. This term incentivises accurate reconstructions. The expectation is taken over the encoder distribution $q_\phi(z|x)$, averaging over the possible latent codes z given the input data x. To maximise this expected log-likelihood, under a continuous Gaussian prior, we equivalently minimise the MSE or cross entropy.

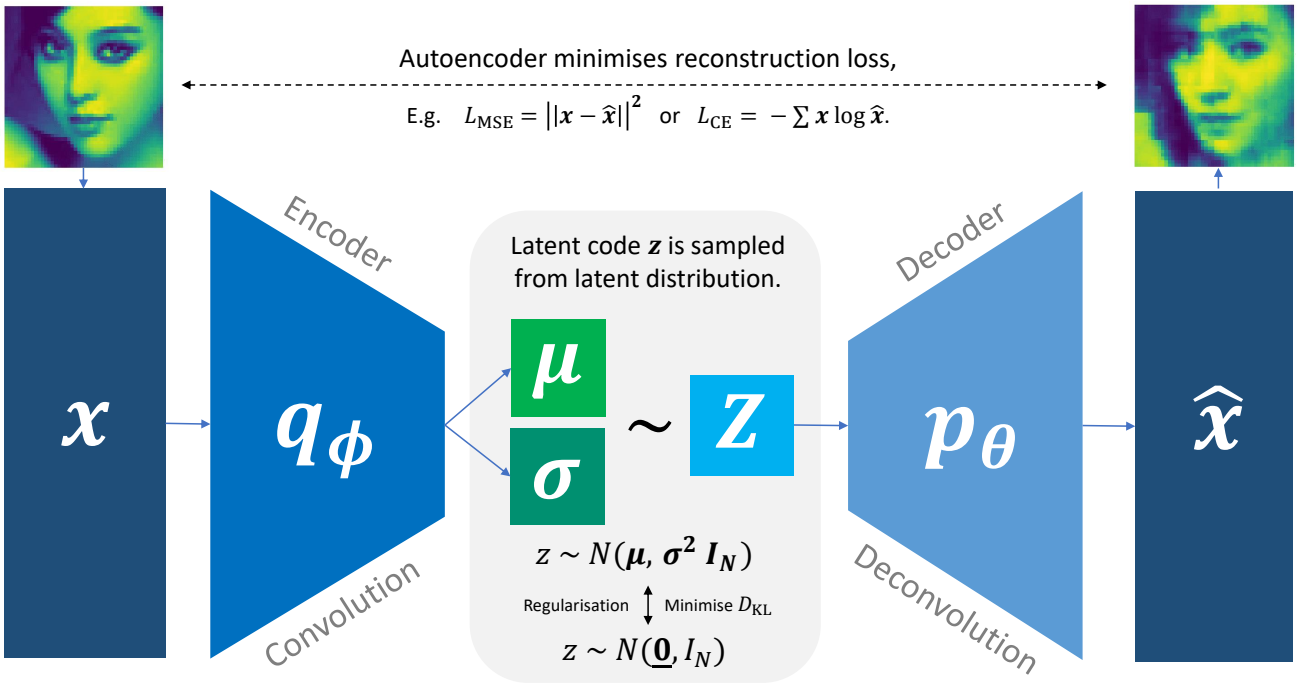


Figure 2.2: The VAE develops the autoencoder by specifying a prior distribution over the latent variables. Here a multivariate Gaussian is used with mean zero and identity covariance matrix.

This approach allows us to use the VAE as a generative model. The trained model can be used to generate new data which is similar to the training data, by drawing samples from the learned latent distribution and passing these through the decoder. In the following sections we apply VAEs to perform image reconstruction, restoration and generation. We explore how the hyperparameters of latent dimension size and beta regularisation coefficient impact the quality of the output images.

2.2.1 VAE Design and Training

In this chapter, the encoders used in the VAEs employ a feature extraction design similar to the one presented in Chapter 1 (as depicted in Figure 1.5), comprising three pairs of convolution and pooling layers. The decoder network is also constructed using three deconvolution layers. However, the capacity of the networks has been enhanced by increasing the width of the layers, leading to a doubling of the number of channels at each output. Similarly, the two prediction networks that interact with the encoder and decoder, respectively, have been widened and augmented with an additional hidden layer due to the complexity of the task. As a result, the models are significantly larger and possess $O(10^7)$ parameters, necessitating longer training periods of 200 epochs. Here, we train four VAE models. The four models initialisations are identical, except that the number of latent dimensions permitted and the value of the beta regularisation coefficient varies.

2.2.2 Test Set Reconstructions

After training, we evaluate the quality of test set reconstructions produced by each of the four models. Fig 2.3, to the right, shows 20 sample reconstructions per model. Like-for-like comparisons are available on the following page in Fig. 2.4. All input images are vertically paired with the reconstructions beneath them.

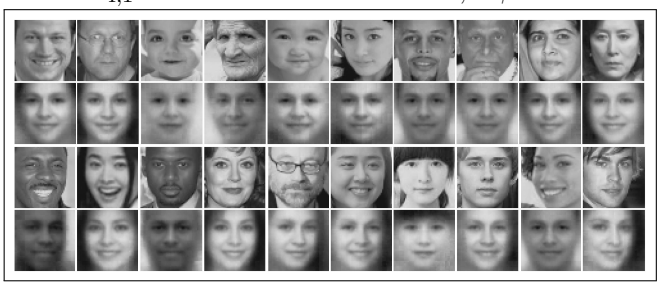
The first model, denoted $M_{4,1}$, has a latent space of 4 dimensions and a β coefficient of 1. The reconstructions generated by this model are largely homogeneous and of poor quality, lacking the ability to accurately capture essential features of the input images, such as facial expression or orientation. These outputs closely resemble the sample means in Fig. 1.2. This suggests that for $\beta = 1$ the loss function (2.1) excessively emphasises regularisation, leading to insufficient reduction of reconstruction loss.

To address this issue, the second model, $M_{4,0.01}$, reduces the regularisation coefficient. The resulting reconstructions exhibit slightly better quality, with a greater willingness to deviate from the prior to improve accuracy. However, underfitting is still evident, potentially due to lack of model capacity.

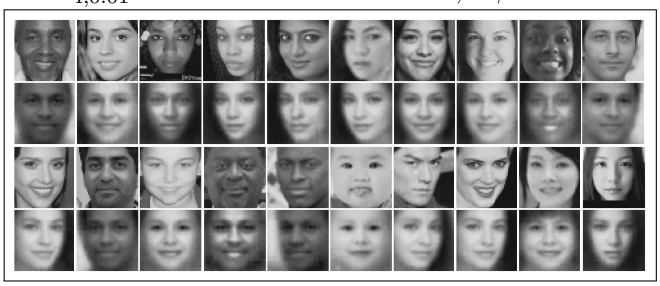
As such, the third model $M_{256,1}$ has been permitted 256 latent dimensions, while maintaining a strong regularisation effect. These reconstructions demonstrate improved visual quality over the previous model, especially in level of detail.

The final model, $M_{256,0.01}$ combines a high latent dimension with a reduced regularisation strength, leading to the highest quality and most accurate reconstructions among the four models. Important details, such as visibility of teeth or facial hair, are captured by this model.

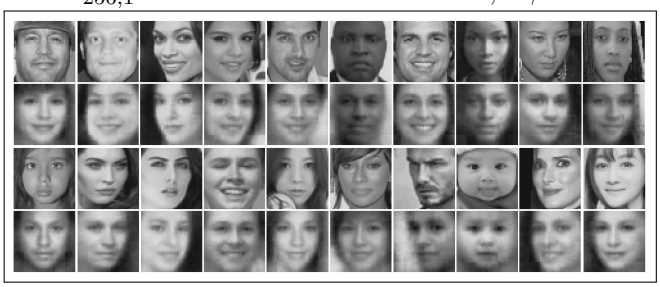
$M_{4,1}$: Latent Dimension = 4, $\beta = 1$.



$M_{4,0.01}$: Latent Dimension = 4, $\beta = 0.01$.



$M_{256,1}$: Latent Dimension = 256, $\beta = 1$



$M_{256,0.01}$: Latent Dimension = 256, $\beta = 0.01$

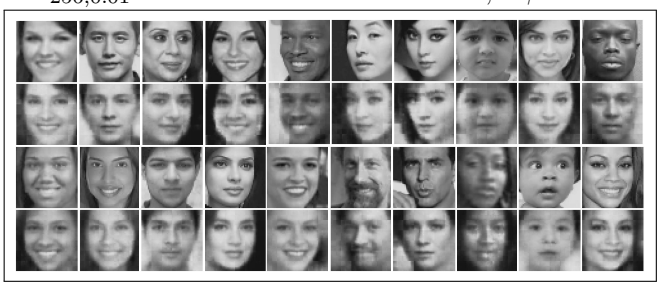


Figure 2.3: Test set reconstructions produced by the four VAE models. Hyperparameter values for each model are shown above each set of samples.

Figure 2.4: Like-for-like comparison of reconstructions applying all four models on the same test samples.

Fig. 2.5 below shows several metrics evaluating the test set reconstructions of each model. These include the peak signal-to-noise ratio, structural similarity index, and mean squared error, averaged over all test observations. The final model $M_{256,0.01}$ demonstrates the best performance across all metrics. For this reason, the following sections will exclusively make use of this model.

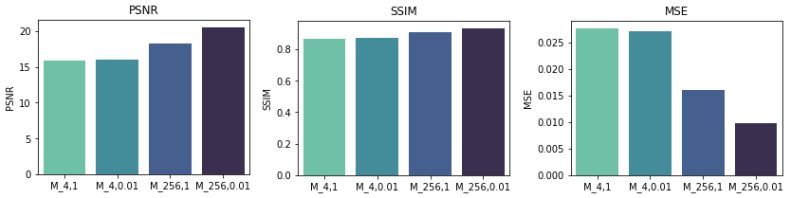


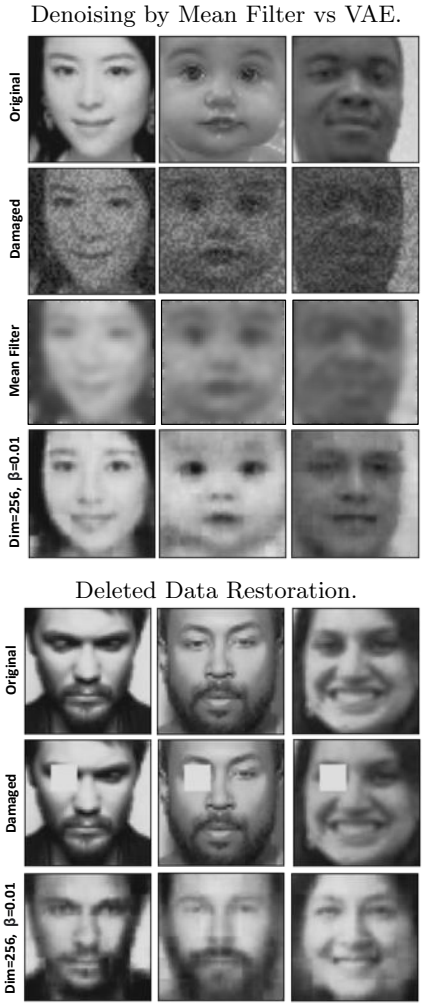
Figure 2.5: Evaluation metrics on test set. Note $M_{256,0.01}$ shows high PSNR, high SSIM, and low MSE, implying best performance.

2.2.3 Damaged Image Denoising and Restoration

VAEs are effective at denoising and restoration, as illustrated in Fig. 2.6, to the right. Here, Gaussian noise has been applied to three images and a subset of data has been erased from a further three images. As can be observed, the $M_{256,0.01}$ model was able to approximate the originals from the damaged images with considerable accuracy. Indeed these VAE reconstructions are of a higher visual quality and sharpness than achieved by the outputs of traditional filter denoising methods, such as the mean filter shown here.

Crucially, the reconstructions generated by VAEs will never be exact replicas of the original undamaged images as they are merely samples drawn from approximated distributions. In this case there are clear non-negligible differences between the original images and the reconstructions, to the extent that one might perceive the reconstruction as being a picture of an entirely different person. Indeed, the reconstruction in the centre column of the deleted data restorations appears to have changed ethnicity, perhaps suitable as a case study in the perils of class imbalance and data bias. Consequently, this approach may be better suited to other domains, such as those involving inanimate objects.

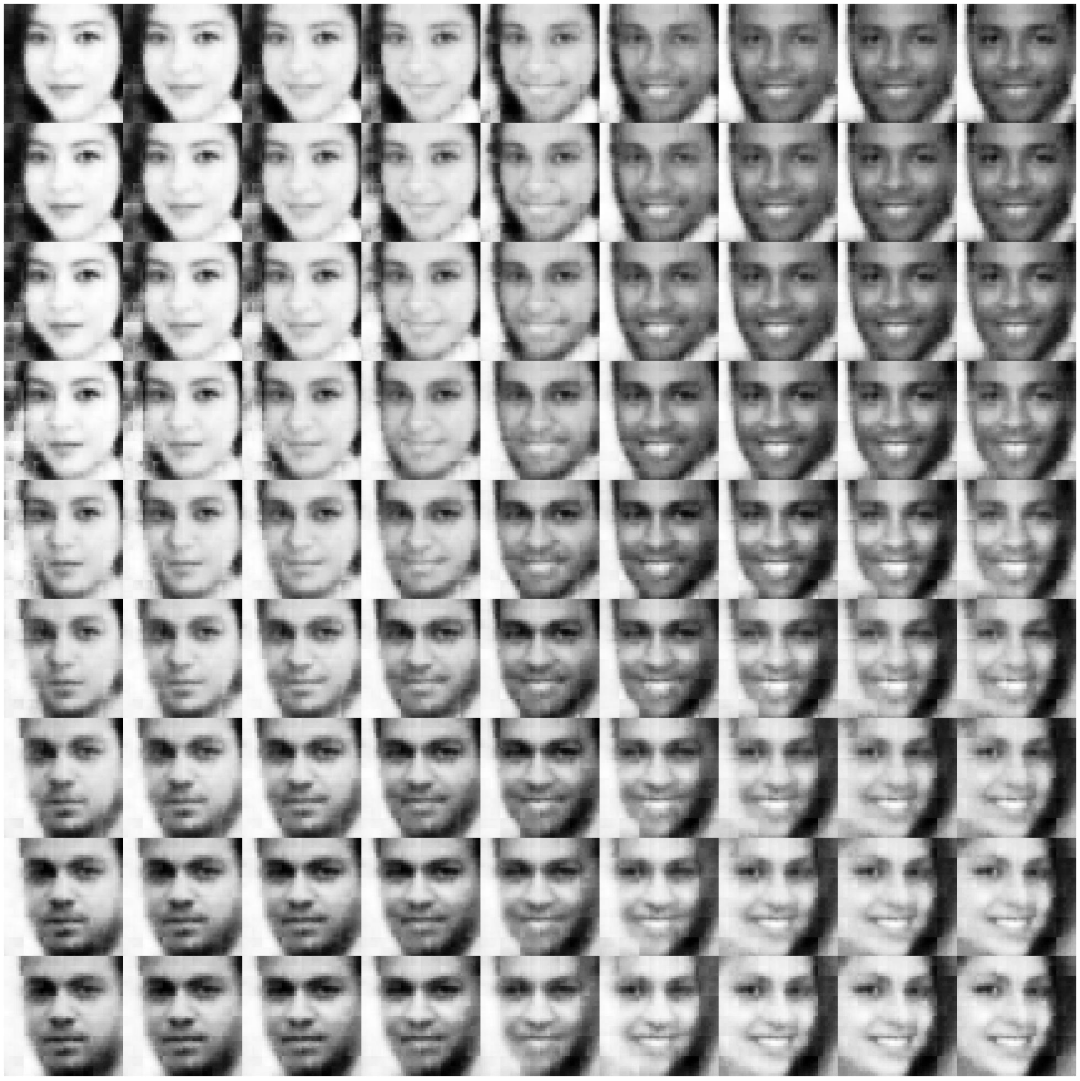
Figure 2.6: The six original images were damaged by introducing noise or deleting data. The damaged inputs were fed into the trained VAE yielding the final reconstructions.



2.2.4 Image Generation and Interpolation

In this section we apply the trained $M_{256,0.01}$ model to generate a face interpolation matrix. Firstly, we generate the four corner faces by decoding samples $\mathcal{Z} = \{z_1, z_2, z_3, z_4\}$ drawn directly from the learned latent distribution. In the latent space, we perform linear interpolation between pairs of these four latent vectors. This is performed by formulating the linear interpolation function, where given two latent vectors z_i and z_j, we have $f(\lambda; z_i, z_j) = (1 - \lambda)z_i + \lambda z_j, \ \ \lambda \in [0, 1]$. We vary λ and build up a collection of latent vectors, taken at regular intervals from along the line segments associated to this function. This collection of latent vectors is decoded and visualised below in Fig. 2.7. Note that no restriction was placed on the initial samples $\mathcal{Z}$, meaning $z_1, ..., z_4$ are not equidistant from one another.

Figure 2.7: Face interpolation matrix. Each face corresponds uniquely to a coordinate in the bounded region of the latent space, whose vertices are the latent codes $\mathcal{Z}$.



2.3 Evaluation

Many reconstructions displayed in this chapter exhibit blurriness or visual artifacts, such as sudden sharp intensity gradients or blocky coloration. This may be due to lack of capacity in the trained models or insufficient training time. With greater computational resources one could train wider and deeper models over more epochs with a higher latent dimension to remedy this, exercising due care to avoid overfitting. Furthermore, the specification of the latent prior is another design decision that governs the output quality and is a rich area of research. In particular, a Gaussian mixture prior may more suitably capture the diversity of classes in the data.

References

1. LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11), pp.2278-2324.
2. Ioffe, S. and Szegedy, C., 2015, June. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In International conference on machine learning (pp. 448-456). PMLR.
3. Kingma, D.P. and Welling, M., 2013. *Auto-encoding variational bayes*. arXiv preprint arXiv:1312.6114.
4. Chou, J., 2019. *Generated loss and augmented training of MNIST VAE*. arXiv preprint arXiv:1904.10937.