Chapitre 3

Study of the HVAE representation of the FFHQ database

In this chapter, we will start by presenting the FFHQ Database and the work environment before presenting the results on the conditional generation.

3.1 FFHQ Datasbase

Flickr-Faces-HQ (FFHQ) [3] is a high-quality image dataset of human faces, originally created as a benchmark for generative adversarial networks (GAN). This dataset consists of 70,000 high-quality PNG images at 1024×1024 resolution and contains considerable variation in terms of age, ethnicity and image background. It also has good coverage of accessories such as eyeglasses, sunglasses, hats, etc.

FIGURE 3.1 – FFHQ dataset [3]

3.2 Deployment environment

3.2.1 PlaFRIM

Experiments presented in this report were carried out using the PlaFRIM tool, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine.

PlaFRIM is a scientific tool based in Bordeaux (France) that aims to support experimental research in all areas of applied mathematics related to modeling and high performance computing. (see https ://www.plafrim.fr).

We used PlaFRIM to experiment with new hardware technologies. We chose to work with the Sirocco machine to have access to powerful GPUs to be able to use the pretrained VD VAE more efficiently.

The Sirocco node characteristics :
CPU : 2x 20-core Cascade Lake Intel Xeon Gold 5218R CPU @ 2.10 GHz (CPU specs).
Memory : 192 GB (4.8GB/core) @ 3200 MT/s.
GPUs : 2 NVIDIA Quadro RTX8000 (48GB).

3.2.2 Python libraries

In this internship, we used PyTorch, an open source machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing, originally developed by Meta AI and now part of the Linux Foundation umbrella. As well as scikit-learn, a powerful machine learning library for Python.

3.3 Pre trained very deep variational autoencoders

During this internship, we used a pretrained VD-VAE network to perform the generation of images [1]. This network was trained on an underscaled version of FFHQ, in 256x256.

FIGURE 3.2 – Examples of imag es generated using VD VAE [1]

3.4 Motivation

The goal of this study is to explore if some semantic information of images is contained within its latent representation.

We used a hierarchical VAE to get the latent representation of the images and hope to generate faces from a specific subcategory (age, gender or glasses for example). We will focus on the gender attribute in this chapter.

3.5 Probabilistic framework

Let x be a random sample from an unknown process $p^*(x)$ that we approximate with $p_\theta(x)$. In this example, x is an image, c the image's class we want to generate accordingly to, and z its representation in the latent space.

- Hypothesis : We suppose that all the information is contained in the latent variables, i.e. :
$$p_\theta(x|z,c) = p_\theta(x|z)$$
Consequently, we have :
$$\begin{aligned} p_\theta(x|c) &= \int p_\theta(x,z|c)dz \\ &= \int p_\theta(x|z,c)p(z|c)dz \\ &= \int p_\theta(x|z)p(z|c)dz \end{aligned}$$

- And using Bayes' theorem,
$$p(z|c) = \frac{p(c|z).p_\theta(z)}{p(c)} \propto p(c|z).p_\theta(z) \text{ with } p(c|z) \text{ our classifier.}$$

3.6 Algorithm

3.6.1 Model

To build our classifier, we chose to use Support Vector Machines because they are effective in high dimensional spaces and are fast when used with linear kernels. In the case of binary classification, the objective is to find the hyperplane that separates the two classes (male and female) while maximizing the distance between the hyperplane and the nearest point from both classes.

After training our model, we can sample each level of latent variables in a way that our samples falls in "the right half" of the plane. We can either :

- Train the classifier on each level and sample them separately :
$$p(z_k|c, z_{<k}) \propto p(z_k|z_{<k})p(c|z_k)$$
- Or train the classifier on every set of levels z<k, freeze the k-1 first levels and sample the next level.
$$p(z_k|c, z_{<k}) \propto p(z_k|z_{<k})p(c|z_{<k})$$

3.6.2 Performance

We split our set of 70.000 latent representation of the images from the FFHQ-256 dataset into 80% for the training set and 20% for the testing set to evaluate the performance of our model.
We start by training our SVM on the k first levels for $k \in [1, 5]$.

As expected, the score increases as we add data :

Latents together	1	2	3	4	5
Dimensions	16	32	288	544	800
SVM score	0.53	0.54	0.66	0.76	0.816

We then train the SVM on each level separately and relate the results :

Latents separate	1	2	3	4	5
Dimension	16*1*1	16*1*1	16*4*4	16*4*4	16*4*4
SVM score	0.5	0.54	0.66	0.74	0.746

We conclude that relevant information on gender is contained in the fourth and fifth level. However, the results aren't as good as we hoped for : for each 100 image generated, we can only hope that 81 of them satisfy our conditions. To boost the performance, we decide to explore different avenues.

3.7 Enhancing the model

There were different hypotheses to explore in order to enhance the model's performance :

- Age lead : The age plays a significant role in our accuracy. The model can distinct more easily between faces of older men and women.
- More levels : We have to use more groups of latent variables for the SVM.
- Non linear classifier : The information is contained within our 5 first groups of latent variables but isn't linearly separable.

3.7.1 Age lead :

We train separately the model on different groups of age expressed in years : [0-6],[7-14],[15-49],[50-120] and plot the validation score :

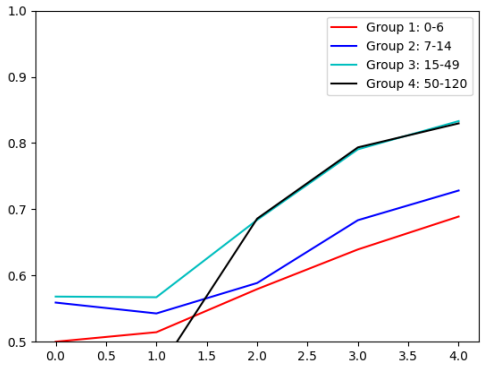

FIGURE 3.3 – Score of the model on different age groups

The model can distinguish more accurately between images of older faces, so we decide to limit our dataset to faces above 15 and train the model.

The score for the model for faces labeled as above 15 is as following :

Latents	1	2	3	4	5
SVM score	0.549	0.547	0.689	0.796	0.843

The performance increases by 3% but is still not sufficient.

3.7.2 More levels :

We use more groups of latent variables as data for the support vector machine. We plot the score for each number of groups of latent variables :

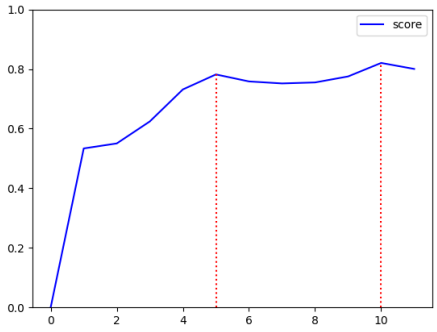

FIGURE 3.4 – Score of the model on levels from 1 to 10

The results shows that the learning rate is stuck at around 80%. That means that adding more level doesn't improve the score of our model.

3.7.3 Multi Layer Perceptron :

In this part, we implemented a multi layer perceptron with two hidden layers and trained it on our dataset.

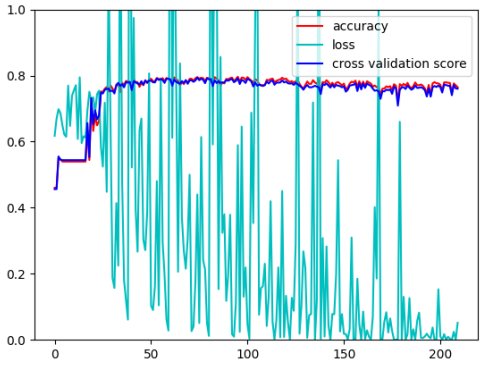

FIGURE 3.5 – Score of the MLP

With this non linear classifier, we obtain almost the same score of 0.8. This means that the model doesn't outperform the SVM with linear kernel.

3.8 Results

After training the classifier, we sampled each group of latent variable, then verified if it falls on the "correct" side of the hyperplane. We then sampled the next group of latent variables while freezing the previous groups. Here are some of our results :

FIGURE 3.6 – Generation of women faces

FIGURE 3.7 – Generation of men faces

3.9 Conclusion

We had hoped to achieve better results. Based on this part, we can conclude that not all the information is contained within the first layers of the latent space. Experiment on age groups had similar results but failed to generalize to glasses since the

The clustering of the hierarchical latent space proposed in this chapter can nevertheless be of interest for image manipulation. This is the object of the next chapter .