Generating Characteristic Faces for varying Spatial Granularity using StyleGAN2

Bart van Laatum
hbpvanlaatum123
@gmail.com

Enrikos Iossifidis
enrikos_i@hotmail.com

Kas Sanderink
kassanderink@gmail.com

1. INTRODUCTION

The goal for this project is to generate so called "Characteristic Faces". These can be understood as faces that typically represent a certain spatial area, based on different levels of granularity (country, continent, world).

The data that is used to generate these faces comes from a Flickr dataset which contains a wide variety of photos (e.g. landscapes, sculptures, family pictures). Section 2 describes how the photos will be analysed, by the use of existing tools. The research questions which this paper attempts to answer are posed in section 3. The dataset and its properties will be discussed at length in section 4. Creating characteristic faces for the each country in the world is challenging, because it requires a database with only "clean" pictures of faces. After the data is cleaned, it must be analyzed to create the characteristic faces for different granularity levels. Lastly, these characteristic faces must be visualized in a static webpage. Besides the photo-related challenges, such as recognizing and analyzing faces, there are 'data size'-related questions of how to efficiently perform operations on millions of photos. In the last section a conclusion is drawn and will be reflected on the posed research questions. The appendix contains an overview of the collaborative efforts made by each team member.

2. RELATED WORK

The main goal of the project is to generate, given the dataset, recognizable and plausible faces for different parts of the world. The analysis of the pictures is divided in two parts, the face detection and face analysis.

2.1 Face detection

The first part of the face detection program needs to be applied to millions of photos, so it is one of the largest tasks to perform during the project. Therefore, a trade off exists between the quality of the face detection and quantity of the photos to be analyzed. The face detection module therefore needs to be simple, but effective. This led to the use of the OpenCV module, which satisfies these requirements. It detects faces for 90-95% when the photo is clear and the photo is taken from the front face, which makes it a reliable choice [4]. How much quality over quantity is wanted can be further specified in the module with the parameters $minNeighbours$ and $minsize$, which represent the quality of the face detection and size of the detected face, respectively.

2.2 Face analysis

For the second part, the face analysis, the StyleGAN2 network from NVIDIA is used. Which is a style-based generative adversarial network developed and trained by NVIDIA [2]. StyleGAN2 is able to reproduce images similar to the images in the training set. For this project a pretrained StyleGAN2 model from NVIDIA is used. The dataset on which the model is trained is the Flickr Faces high-quality (FFHQ) dataset[1], which contains 70K photos of faces from all over the world. The FFHQ set contains images of faces with some range in diversity in terms of ethnicity and age as well. However, it is questionable how well this set is a robust representation of the world population. The process of training the model and generating new images is structured as follows:

Training network:

1. Neural network takes a photo as input and as an output a vector in the latent space.

2. Neural network's weights are trained with the FFHQ dataset.

After the training phase, the model has created a latent space in which each axis represents a face feature. The pretrained model is able to generate new images of faces using the input of some latent vector. This works as follows:

Generate new photos:

1. Generate a random vector.

2. The vector is embedded into the trained latent space.

3. The obtained latent vector is transformed to a photo, which gives the newly generated photo .

Another functionality of StyleGAN2 is to map an image to a vector in the latent space of the model. Latent vectors represent images in a compressed and simplified form. The latent vectors of images that show more resemblance

Proceedings of the VLDB Endowment, Vol. x, No. xxx
ISSN 2150-8097.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx

[1] `https://github.com/NVlabs/ffhq-dataset`

between each other, tend to lie closer to one another in the latent space. Obtaining the latent vector of a photo is not universal and model dependent. StyleGAN2 provides a tool to approximate the latent vector of an image. The model does so by applying gradient descent optimization on a vector to approximate the latent vector of the real image as close as possible. Based on this approximated latent vector, StyleGAN2 is able to generate an image that resembles the real image as close as possible.

Figure 1 visualises the process of approximating the latent vector of a real image. In the left the picture of an authentic Dutch boy is shown. Next, through gradient descent optimization, the latent vector z' is approximated. When feeding this z' to StyleGAN2 a image is generated that resembles the real picture as close as possible.

As figure 1 visualises, latent vectors represent images in the latent space of the StyleGAN2 model. This indicates that taking the average over several latent vectors corresponds to the average of the images that are represented by specific latent vectors. Therefore, the latent vector of each picture shall be approximated. Next, the mean latent vector per region is used as input to generate an average image using StyleGAN2. Section 4.2 will explain the process of generating characteristic faces of the world from latent vector in detail.

3. RESEARCH QUESTIONS

To generate the characteristic faces, the following three main questions will be researched during the project:

1. Main question: Does averaging latent vectors generate characteristic faces?
2. What is the minimal amount faces necessary to generate characteristic faces?
3. What is a characteristic face?

The main question is related to the methodology used during the project to generate the characteristic faces. It therefore tries to answer if the choices made during the data preparation and the implementation of the models from section 2, eventually lead to reaching the project's goal.

The second question tries to answer the project's practical limitations, because there is a limited amount of pictures that can be prepared and analyzed. As a consequence, there is a trade off between the quality of each generated characteristic face and the quantity of the faces that must be generated for all countries, continents and the world.

The third question tries to answer the social and biological aspects of what humans perceive as a characteristic face for a country or continent how this perception is created.

4. PROJECT SETUP

4.1 Data preparation

In this section the raw data is first described as well as each step in the data preparation. Additionally, the data characteristics at each step are given and the data distributions of the final data product.

The given raw dataset contains 10 bz2-files. Each file is little over 1 GB in size, bringing the size of the total dataset to 12.81 GB. These files were unzipped with the *%sh zip* magic command and loaded as 10 separate parquet-files into Databricks. During the whole project, the files were saved as parquet-files. The files were converted to RDD objects when the data was being analyzed. To perform the analysis, the map-and-reduce method and in-built Databricks functions were mainly used.

In Figure 2 the pipe line of the data, from input to output, is shown. Table 1 shows the size of the data for each intermediate step, as well as an approximation of the computation time necessary to reach it. Both figure 2 and 1 can be used as a reference throughout this section.

First, the raw data. It roughly contains 100 million lines, i.e. possible pictures. Every line contains a string with a reference to the Flickr page and tags, such as the latitude and longitude. The total size of the data is 12.8 GB, which gives an average of 128 bytes per line.

In the first step, each string is parsed and checked for a latitude and longitude and reference to Flickr. The second step is testing whether the strings with a location have a valid Flickr reference by trying to load the picture in the link. If a picture can be retrieved from Flickr, then the bytestring of the retrieved picture is saved. Additionally, the corresponding country and continent of each latitude and longitude is retrieved with the Python modules *reverse_geocoder* and *pycountry_convert*.

In the third step, the face detection is performed on the loadable and correctly tagged pictures, using the OpenCV module. If so, the picture is saved. The filtered data is saved in separate parquet files. Each row includes the bytestring of the picture, the country and the continent.

The steps were performed on the first 6 given bz2-files and resulted in a total of 437.808 rows, a parquet-file of 25.91 GB and an average of about 60 kilobytes per row.

Next, in the fourth step, the pictures are aligned, such that they can be optimally embedded in the latent space of the StyleGAN2 model. It is a function coming from the StyleGAN2 model. The aligning process includes another face detection and reshaping and resizing the whole picture. Furthermore, adjustments are made to certain parts of face, e.g. the eyebrows or jaw, such that the face can be better embedded by the StyleGAN2 model.

This finally results in 322.278 aligned pictures, ready to be used by the StyleGAN2 projector. Since the pictures are resized and improved, the size of each picture is significantly increased. As a result, the size of the parquet-files grows to 302.67 GB with an average of 940 kilobytes per instance row.

To keep the run time of the StyleGAN2 model within the temporal and financial limits of this project, the number of faces per country is limited to 25 faces. Additionally, the chosen world map visualization contains 173 countries. Therefore, after limiting the number of faces to 25, only countries that were included in the web visualization were saved. The limited dataset contains 3758 rows, a total size of 5.73 GB and about 1500 kilobytes per instance row.

The last step, before creating the characteristic faces, is categorizing the data in terms of gender and age. This is performed by using the OpenCV module again. It returns man or female for the gender detection and 8 different age categories for the age detection. After categorizing, the data is ready to generate characteristic faces.

4.1.1 Properties Data Product

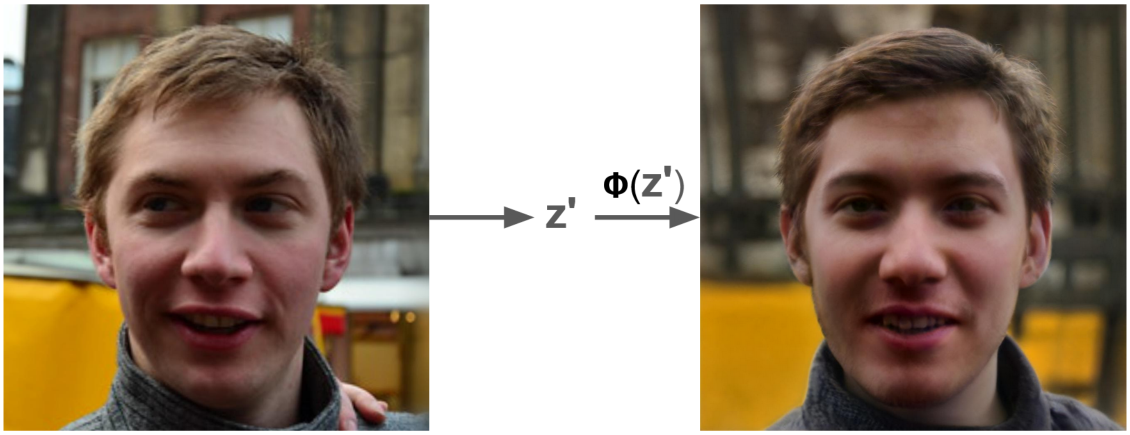

Figure 1: The approximation of a latent vector visualized. By applying gradient descent optimisation latent vector z' of the is obtained. Which depicts StyleGAN2's representation of the left image. When feeding this z' to StyleGAN2's generator function generates a picture resembling the real image as close as possible.

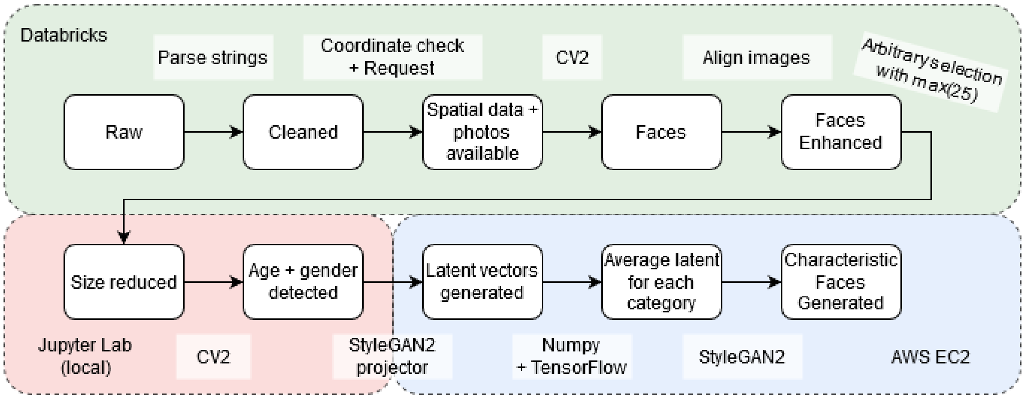

Figure 2: The data processing pipeline. The white boxes with black outlines describe the states of the intermediate data product. The white boxes without outlines at the top and the bottom describe the performed transformation between two states. The three colored areas with dashed outlines indicate the platform on which the transformations were done. Green, red and blue stand for Databricks, Jupyter Lab and AWS EC2 respectively (as indicated in the figure). Each stage is described in varying detail based on the relevancy in section 4.1 (green and red area) and 4.2 (blue area).

Stage:	**Size** (GB)	**Instances**	**Ctime**
Raw	12.81	100M	-
Cleaned & Spatial data + photos available (6 of 10)	382.61	5.43M	35h
Faces (6 of 10)	25.91	437K	10h
Faces Enhanced (6 of 10)	302.67	322K	8h
Size reduced	5.73	3758	1h
Age + gender detected	5.73	3758	3h
Latent vectors generated & Average latent for each category	2.7	3758	42h
Characteristic Faces (png)	1.14	1593	3h

Table 1: For each intermediate data product, this table shows: the size, the number of instances, and the computation time needed to generate it. The steps were not performed on all 10 files, which is denoted if necessary. In two cases, two steps from figure 2 are combined into one row because no intermediate data product was saved. This is indicated by the &-sign. All numbers displayed in this table are estimations, and should be interpreted as an indication of the order of magnitude.

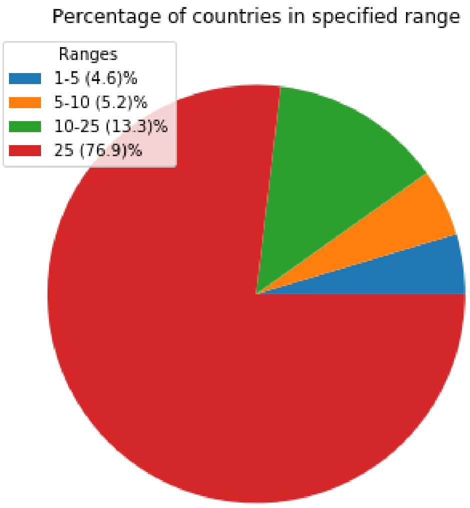

Figure 3: Pie chart of the data in web visualization. The size of each range depends on the number of countries that have the number of available aligned faces in that defined range.

Before StyleGAN2 is applied to the data product, the data´s properties are first discussed.

Figure 3 shows the distribution of the number of faces per country that were found after detecting, aligning and maximizing the number of faces for the 173 world map countries.

It shows for example that for 133 (76.9%) countries, at least 25 aligned faces were found from the raw data set and for 8 countries, only '1-5' aligned faces were found. As a consequence, the average 'characteristic face' will be influenced more by the available faces as observed in the result section.

The categorizing step on the aligned and the randomly selected 25 faces (or less), returns Figure 4 as a result. Besides the number of faces available, the average face is also affected by the gender and age of the available faces.

For man and woman, the ratios are roughly equal over all age categories. In contrary, the selected aligned faces are mainly contained in the age categories '8-12' and '15-20'. This can be explained by the random selection of faces. Another explanation could be that certain age categories are over-represented in the raw data set.

4.2 Application of StyleGAN2

In this section the use of the StyleGAN2 model to generate characteristic faces will be discussed.

First, the process of retrieving latent vector of all images is described. Next, will be explained what kind of Amazon Cloud Computing service is used in order to obtain these vectors. Last, is described how the objective of generating characteristic faces is accomplished.

4.2.1 Obtain latent vectors

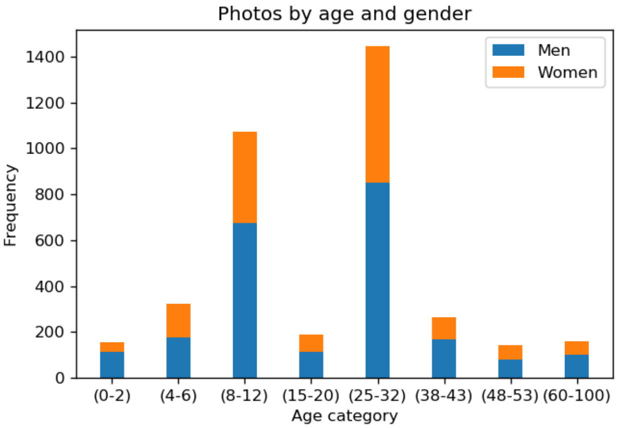

Figure 4: Stacked histogram that represents the frequency of each age category in the final data product. Additionally, each age category shows the proportion of man and woman.

After the data preparation the images are embedded in the latent space of StyleGAN2 as described in section 2.2. For all images in the data product, the latent vector is approximated by StyleGAN2's approximation tool. To optimise the performance and the run-time of this approximation, Amazon Cloud Computing services is used. Next, will be discussed what kind of services are used and why.

4.3 Amazon Cloud Computing

StyleGAN2 has strict requirements on the versions of the software it uses. The model is only compatible with Python version 3.6 or 3.7. Moreover, StyleGAN2 was build using Tensorflow 1.x, therefore Tensorflow 2.x is not supported. Also, the model cannot be used without a GPU core of at least 12GB [1]. Therefore, the P3.2xlarge Amazon EC2 instance is used to meet the required hardware and software restrictions. P3.2xlarge instance consists of a single NVIDIA Tesla 100V GPU, with 16GB of GPU memory. This instance is chosen since the pre-trained model used the same hardware. Using this EC2 instance gave a speed-up by factor 20, when calculating the latent vectors. However, obtaining the vector for an image still takes up to 36 seconds. Therefore, mapping images to the latent space remains a costly computational task.

4.3.1 Generate Characteristic Faces

After approximating the latent vectors for all the images in the data product, the characteristic faces can be generated. In order to do so, the average of all latent vectors from a region is taken. This returns mean vector that represents a specific region. When StyleGAN2 generates a picture from this mean vector the image of face is produced in the same manner as visualised in figure 1. Besides taking the average for all latent vectors from one region, the gender and age classification make it possible to generate averaged faces upon specific queries. For each region a male and a female face is generated by averaging the latent vectors of a specific gender. Similarly, this is done for latent vectors corresponding to young and old classified faces. As well as

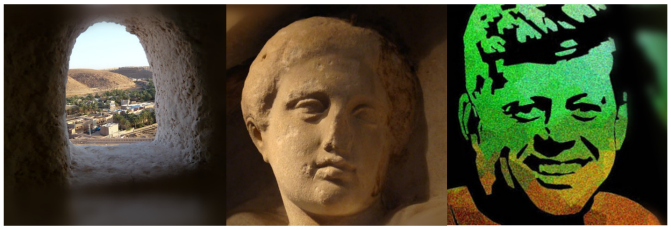

Figure 5: Three examples of images that do not contain a human face that neither of the face-detection algorithms filtered out. A sculpture and a painting are shown, as well as photo of a valley through a small window.

for all combinations of these gender and age classifications. In the results section generated faces for each specific query will be displayed.

4.4 Discussion

In this section, some possible points of improvement will be discussed.

First off, that a picture is taken in a certain country, does by no means entail that the person(s) in the picture are representative of the country. The prime example of this is vacation photos. This was not taken into account for during the processing of the data. In a subset of the raw strings in the original data, photo tags are provided. A possible measure to filter out vacation photos is to check these tags for words associated with vacation (e.g. "travel", "holiday").

Secondly, there are photos in the data product that do not contain faces, despite the usage of two face detection algorithms. These photos typically contain sculptures or paintings of faces, but not necessarily (see figure 5). These photos could be filtered out manually.

Thirdly, the StyleGAN2 model is trained on the ffhq dataset. This puts a limitation on the range of faces that could possibly be generated. This is illustrated by the fact that the average face of the world is the average face of the model. Thus, inescapably, the diversity of the generated faces is bounded by the diversity of the StyleGAN2 model.

Fourthly, the gender and age detection model seems to perform worse for people of color. This can be explained by a training bias. The model was possibly trained on a dataset in which people of color were underrepresented.

4.5 Results

The results can best be explored through the web application. However, a small sample of the generated characteristic faces will be shown here. Figure 6 shows four faces from four different countries. The photos are chosen to show the impact of the number of available source photos on the generated image. For Suriname, only 3 images were available. Evidently, these photos did not contain typical faces, giving in a non-representative result. The sample size for the second image - a characteristic face from Mauritania - was also relatively small, with only 7 photos. Due to this the face is somewhat deformed. The last two images - faces from Malawi and Russia generated using 18 and 25 photos respectively - look significant better (admittedly, it is hard to quantify what "better" means in this context).

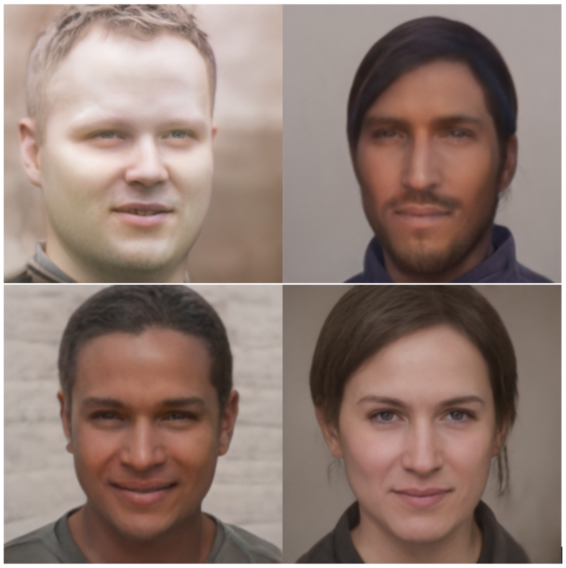

Figure 6: Four faces from four different countries with varying amounts of available source photos. Each category defined in figure 3 is represented by one image. Top left: Suriname, 3 photos. Top right: Mauritania, 7 photos. Bottom left: Malawi, 18 photos. Bottom right: Russia, 25 photos.

4.6 Visualisation

4.6.1 Front-end

The goal of the visualisation is to be simple and intuitive. The visualisation consists of two components: an interactive world map on the left and a dashboard on the right. The dashboard is used to display the generated images, and to control the parameters (i.e. gender, age and locality) via a form. When a country on the map is clicked, a photo is displayed using the selected parameters. When a parameter is changed, the photo that is displayed changes as well. The number of original photos available in the selected country is displayed along with the image.

4.6.2 Back-end

The original implementation of the map (published in a Github repository [3]) had nearly all the functionality needed for the application. When a country is clicked, the iso-code of the country is logged.

Two events can cause the displayed photo to change: a different country is clicked, or a different parameter configuration is chosen. Both events trigger the same course actions. The information provided by a combination of the form and the world map is encoded. For example: A request for a young (Y) male (M) from the Netherlands (NL) is encoded as NLMY. This same encoding is used as a name for the corresponding image stored in the `/static` folder, making the display of the photo as simple as changing the source of a dedicated HTML img-object to `src=/static/NLMY.png`.

5. CONCLUSION

The project's goal was to generate characteristic faces for all countries, continents and the world.

To start off, the third research question was answered by defining a characteristic face as the average latent vector of a country, continent, or the world - where the faces of each country are picked randomly. As a consequence, some age categories were over-represented, which influences the average latent vector. This results in a bias towards the age category 25-30. Picking the faces according to a country's population composition or picking the faces uniformly could give a more representative image.

In order to answer the second research question, the characteristic faces of countries with different numbers of source photos were compared. The quality of the average face highly depends on the quality of the selected source photos. Therefore, even with a small number faces, such as '5-10', a reasonable characteristic face can - in some cases - be generated.

Finally, the main research question is answered by generating the average faces of all countries, continents and the world in the visualization.

During the data preparation the size of the data was significantly increased in the steps of saving the photos and aligning the faces. Additionally, because StyleGAN2 is not compatible with DataBricks, the usage of a separate AWS EC2 machine was required in order to produce the images needed for the final visualization.

Nevertheless, the world map visualization shows a face which is recognizable as characteristic for that country or continent, for the majority of the countries. The StyleGAN2 model generates, even with a low number of faces, a new, realistic face. Thus, it can be concluded that StyleGAN2 is a tool capable of generating characteristic faces. It should be noted however that there is room for improvement. Some possible areas of improvement have been identified in the report, the most notable one being the noise generated by vacation photos.

6. REFERENCES

[1] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In Proc. NeurIPS, 2020.

[2] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020.

[3] Mustafa Omar. Jsvectormap. `https://github.com/themustafaomar/jsvectormap`, 2020.

[4] Philipp Wagner. Face recognition with opencv. Order A J. Theory Ordered Sets Its Appl, pages 1–26, 2012.

Project part:	**Bart**	**Enrikos**	**Kas**
Report			
High involvement		X	
Medium involvement	X		
Low involvement			X
Code			
High involvement	X		
Medium involvement			X
Low involvement		X	
Visualizations			
High involvement			X
Medium involvement		X	
Low involvement	X		

Table 2: Division of tasks during project

7. APPENDIX

Table 2 shows a rough estimation of the distribution of the collaborative efforts made by all three team members.