PoseBias: On Dataset Bias and Task Difficulty - Is there an Optimal Camera Position for Facial Image Analysis?

Mohit Choithwani
mohit.choithwani@fau.de

Sneha Almeida
sneha.almeida@fau.de

Bernhard Egger
bernhard.egger@fau.de

Cognitive Computer Vision Lab (CogCoVi)
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

Abstract

Let's imagine you could choose the position of the camera for a particular face analysis task - where would you put it? In this work, we provide a first analysis based on synthetic training data to provide evidence that this choice is not trivial, not only dependent on the training data and different based on the task. We provide results for two major face analysis tasks, face recognition and landmark detection. For our experiments, we use a 3D Morphable Model as it provides us full control over pose, illumination, and identity to generate idealized training data. Whilst rendered images are not photorealistic we avoid any confounding factors and biases from other sources (e.g. pose bias in training data).

Our results show that the optimal camera poses are near frontal but not exactly frontal and dependent on the task. By comparing the results obtained by pose-specific training set to a uniform training distribution without pose bias we show that the accuracy for both tasks not only depends on the bias in the training data but is actually dominated by the difficulty of the task depending on the particular pose.

1. Introduction

Face analysis tasks are becoming more and more impactful in our daily lives. For example entertainment systems might track some facial landmarks to control games and track the players mood or cars might recognize the state of the driver or their identity. For all those systems we mount cameras and the position is determined by several different constraints, like where they are not in the way - we propose to not only choose the position of the camera by practicality but also by the expected performance for the face analysis tasks based on the camera position. Capturing the face from a position that maximizes the number of features extracted from the face is the key to performing such tasks on faces. We attempt to find an optimal position for the camera in a specific setup or e.g. inside the car. We further aim to find if this optimal position changes according to the task, or remains the same, irrespective of the task.

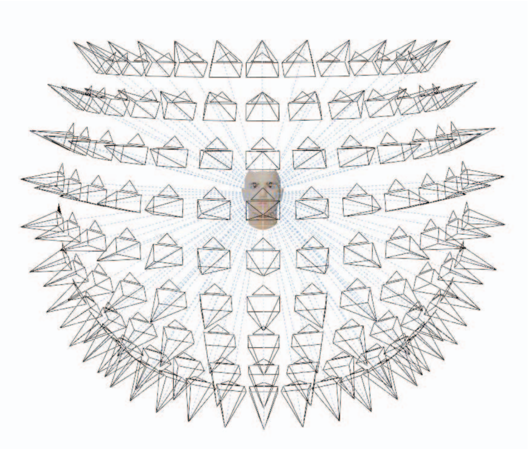

Figure 1. Imagine you could choose one of those 117 cameras for a specific face analysis task - which one would lead to the highest accuracy? This is the core question we aim at in this paper.

To make these findings, we perform a series of experiments on synthetically generated human faces. We mainly focus on two tasks, face recognition (which is an application of the supervised learning technique, classification) and facial landmark detection (which is an example of regression), where we consider faces from different camera angles and positions. We first find the distribution of the performance based on the camera position for the face recognition task, and then for the facial landmarks detection task. Then we compare how these two results are different and how the optimal camera positions to capture a face differs according to the task at hand.

As both the tasks involved here are supervised learning tasks, labeled data is needed. Labels are different for both, classification and regression. For classification, labels are the identities associated with each face that the classifier has to identify. For regression, labels are the landmark positions that have to be estimated.

To summarize, the main contributions of this work are the following:

- We analyze the difficulty of a task in relation to camera pose separated from any other bias in datasets.
- Our results indicate that there optimal camera positions that is not necessarily frontal.
- Our results suggest that the optimal camera position might be dependent on the task at hand.
- We demonstrate that the dominating factor for the reached accuracy is the task difficulty, rather than the bias in the dataset.

The findings of these experiments can be used as a foundation to build applications that will enrich the user's experience based on face tracking or feature extraction. Ultimately this for example might enhance the safety of a driver in a vehicle through more reliable detecting emotions or drowsiness from the driver's face. Or detecting if the driver is falling asleep and activating an alarm to alert the driver or using the driver's face for unlocking the car instead of a key.

1.1. Related Work

We are using synthetically generated faces for face recognition and facial landmark detection to find if the amount of face visible to the camera influences the difficulty of the task. This mainly depends on the position of the mounted camera as the angle determines the degree of self-occlusion. [11] performs similar experiments on synthetically generated faces to find if the performance of Deep Convolution Neural Networks (DCNNs) is influenced by data set distribution like illumination conditions and pose variations. They also investigate how well a DCNN is able to generalize when different identities do not share the same pose variation. They also demonstrate that their findings on synthetic data also can be transferred to real-world data. This work is an instance of measuring the bias in a dataset with synthetic data that is generated in a similar way as ours. In contrast, we do explore the bias in performance caused by a shift of the presented distribution (rotation).

In [13], experiments have been performed to test the performance of deep face recognition systems on synthetic data, real-world data, and combinations of real-world data with synthetic data. It has been concluded that deep face recognition systems benefit from synthetic training data and there are no observable negative effects of pre-training with synthetic data. Moreover, the use of synthetic data can significantly reduce the amount of training data needed to train these systems. [12] establishes that synthetic data has the potential to reduce the effects of biases that deep face recognition systems face because of real-world training datasets. Combining synthetic data with real-world data in the training phase helped reduce the size of the real-world training dataset by 75 percent while still maintaining the competitive performance of the face recognition architecture. Also, synthetically generated datasets were fully annotated, which made it possible to evaluate the generalization ability of neural network architectures. Those two works use similar synthetic data as we do but mainly in overcoming dataset bias.

[11], [13], [12] worked with synthetically generated facial data to study and overcome bias. They proposed the generator we are using here and asked specific questions about dataset bias by modifying the data distribution in a controlled way. They observed that some results obtained on synthetic data can be transferred reasonably well to real-world applications. They propose, that synthetic data has a lot of potential in training neural network architectures to the effects of dataset bias. Methods to measure and mitigate bias with synthetic data are further surveyed by [22] focusing on bias through demographics.

Although Generative Adversarial Networks (GANs) have been shown to be effective at synthesizing high-quality facial images [15] [16], there are several limitations associated with GANs used for image generation. One major limitation is that GANs are often not fully controllable [18], making it difficult to specify the exact attributes of the generated images. For example, generating images with large pose deviations may not produce accurate results, and the level of control over camera positions is limited [20]. Another limitation is attribute entanglement [10] [9], which occurs when the generator learns to encode multiple attributes in one or a few latent variables. As a result, changing one attribute of a generated image may indirectly change other attributes, making it difficult to control or modify specific attributes of the generated image. GANs are also highly dependent on the quality of the data on which they were trained. If there is some bias in the training data, the generated images may reflect those biases. Finally, GANs can be computationally expensive to train, and generating high-quality images can require significant computational resources. The main reason why we suggest not using them in our context is the missing guarantees in terms of the multi-view consistency of such methods. There is a recent dataset called Syn-YawPitch [7] proposing to use 3D GAN derived data to study pose-related bias in face recognition. They observe a strong bias towards frontal poses which is from our perspectly caused not only by the face recognition models at hand, but as discussed also based on the frontal bias in the training data for the 3D GAN generator network.

In addition to those works in the field of Computer Vision and Generative Modelling there are also interesting connections to studies aiming to understand human vision. As reported decades ago, the human visual system performs better for certain poses than others. In particular a so called three quarter view ($45°$) was shown to lead to better face recognition performance than a frontal view [2, 14, 17, 21]. This is an interesting parallel worth exploring further.

2. Methods

Our methods consist of three core components: data generation via a parametric image generator, training of a task-specific face analysis network on parts of the generated data, and finally the evaluation of the resulting network. An overview of our approach is presented in Figure 2 and in the following, we describe those components in more detail.

2.1. Camera Specific Image Generation

For generation of our training datasets, we rely on a 3D Morphable Model [1, 5], namely the publicly available Basel Face Model 2017 [6]. This model is a statistical model for shape and color and enables explicit control over identity, expression, pose, and illumination. Whilst our data generation tool does not lead to photorealistic images we can guarantee multi-view consistency and all controllable parameters are by design disentangled. For our investigation, we are mainly interested in controlling the pose and we therefore randomly sample from a specific distribution of poses. For each of the camera positions we investigate, we sample an individual dataset. For this data generation process, we are using the parametric-face-image-generator, an open-source framework that was used in prior work to measure and mitigate bias in face recognition systems [11, 13, 12]. This face generator generates synthetic faces of different identities and can add expressions. The resulting 3D mesh is then rendered under a pose and illumination condition sampled from a predefined distribution. This enables us to generate an arbitrary amount of 2D face images with labels for a desired distribution (here in particular the pose distribution). We describe the used distributions and parameters in Section 3.1. The pose is parameterized via pitch, yaw, and roll as well as translation. We keep everything besides yaw and pitch fixed to focus on the effect of the viewing direction, see Figure 3. The illumination condition is parameterized through Spherical Harmonics and we deploy the Basel Illumination Prior as a distribution of natural illumination conditions.

2.2. Camera Position Specific Training

During data generation, we keep all distributions constant except for the pose distribution. By only changing pitch and yaw for each camera position we generate datasets that have all the same distribution besides the different viewpoints. This enables us to study the effects of different camera positions in isolation from all other confounding factors (e.g. pose dependent effects on learned models like GANs due to pose biases in training data). After generating training data for performing the said tasks for different camera positions, we train and evaluate a model based on these subsets to evaluate the task difficulty based on the generated images. For each camera position, we train a separate model and we do so for the task of face recognition as well as landmark detection.

2.3. Camera Position Specific Evaluation

The view captured by the camera largely depends on the placement/position of the camera. When we consider a specific case of capturing the human face through the camera, different positions of the camera capture it from different angles, thus capturing different features of the face. We evaluate training and testing accuracy/loss and study if and how they depend on the selected pose. More details about this analysis and especially the selected pose distributions are covered in Section 3

The method at hand and our experiments following this setup have been designed to find satisfactory answers to the following questions:

- Is there an optimal camera position from where facial features can be better extracted?
- Does this optimal position differ according to tasks, e.g., can we have different optimal camera positions for recognition and landmark detection?
- How important is the camera position in contrast to the training data distribution? Does the difficulty of the tasks differ for different poses?

2.4. Measure Task-Specific Bias

To study the effect of the camera position on the performance of networks we need to fix a task for training and evaluation. This enables us to see if the bias is depending on the task at hand solely, or if the measured effect is also caused by the data distribution. We investigate two particular tasks, namely face recognition and landmark detection.

2.4.1 Face Recognition

This is an image classification task. Here we train a neural network to identify a face. So, the input to the neural network is a set of facial images and their associated identities.

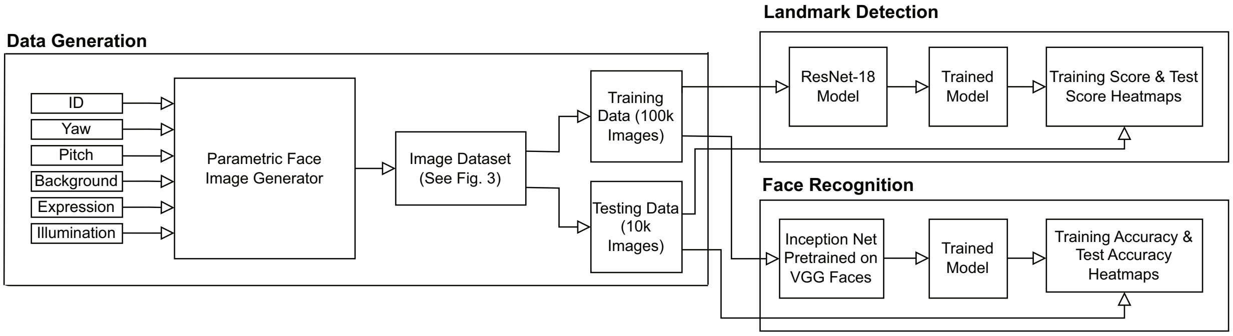
Data Generation
ID
Yaw
Pitch
Background
Expression
Illumination
Parametric Face Image Generator
Image Dataset (See Fig. 3)
Training Data (100k Images)
Testing Data (10k Images)
Landmark Detection
ResNet-18 Model
Trained Model
Training Score & Test Score Heatmaps
Face Recognition
Inception Net Pretrained on VGG Faces
Trained Model
Training Accuracy & Test Accuracy Heatmaps

Figure 2. Overview of our data generation and training approach.

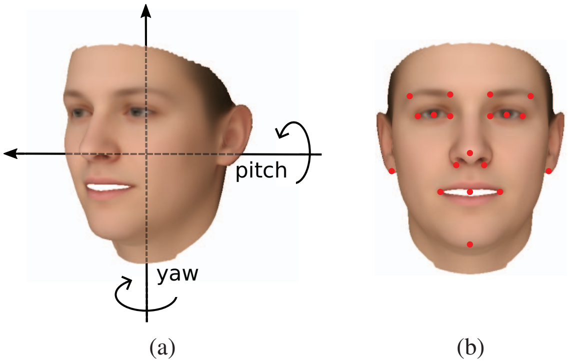

Figure 3. We rotate around vertical and horizontal axes to get pitch and yaw variation (arrows point in negative rotation direction) (a). The 19 facial landmarks used for our landmark detection task (b).

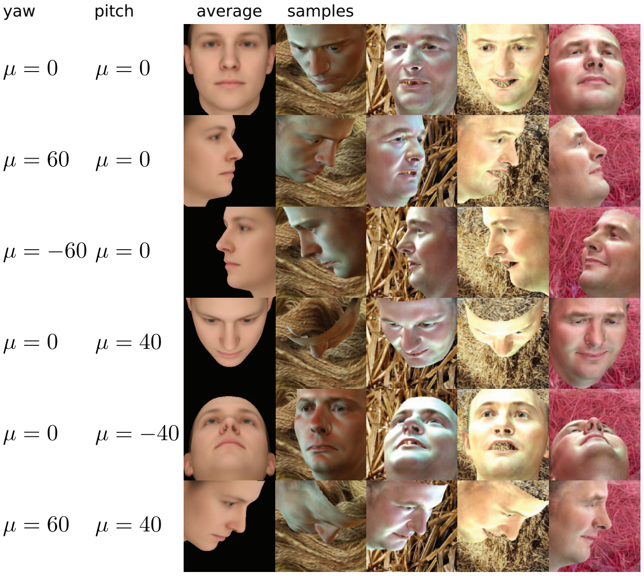

Figure 4. Data generation by varying yaw and pitch explained with images from the generator variance $\sigma = 20$ for all **Note:** All images are generated from the same identity varying expression, illumination and pose. Images in the first column are for visualization of the mean pose only, they do not have any variance and are not used during training.

The neural network learns the mapping between those images and their identities. Then, it can identify the identity of the image when only the facial image is provided (given that the image belongs to one of the identities on which the neural network has been trained). Face Recognition is a supervised image classification task that requires labeled data. In this case, data consists of images of the face and labels are the identifiers associated with each image. The 3DMM creates this data, which has facial images and associated identities. Next, we need to train a model which can operate as a face recognizer. This model should be able to take a facial image as input and predict its identity. We create this model using a pre-trained neural network. We use Inception net ([19]) that has been pre-trained on the VGG faces dataset ([3]) and use the technique of transfer learning for training it on our data. We adapted the architecture for this task by removing the last layer and replacing it with a layer for 1000 classes/identities. We trained the model for 20 epochs with a batch size of 32 and used an Adam optimizer with a learning rate of 0.001.

2.4.2 Facial Landmark Detection

This is a regression task performed on facial landmark points. We have considered 19 landmarks on the face for this purpose. Here we train a neural network to identify landmark positions on the face. The input is the set of facial images and landmarks associated with each face. As in the previous task, the network learns the mapping between faces and their landmarks. So, when we provide the image of a face as input, we receive the landmark positions of the face. Facial Landmark Detection is a regression task and we train it in a fully supervised fashion. The data consists of facial images and the labels are locations of landmarks associated with each facial image. Then, we create a model that is capable of taking a face image as an input and detecting the location of landmarks in the output. We use ResNet-18 ([8]) and train this model on our data. We removed the last layer and added one with $3 \cdot 19 = 57$ nodes and trained this new layer from scratch. We trained the model for 20 epochs with a batch size of 32 and used an Adam optimizer with a learning rate of 0.001.

3. Experiments

We perform four different experiments to study the effect of the camera position on downstream applications. For all experiments, we keep the data distribution constant but shift it by rotating and moving the camera in 3D space. Our first two experiments train and evaluate a network for all camera positions we consider for the task of landmark detection and face recognition. Our third experiment measures the bias we would observe with a perfectly uniform data distribution. If the difficulty of the task is strongly depending on the position of the camera we should see this effect in all our three experiments. The fourth experiment investigates how reliable our experiments are in terms of noise.

3.1. Data Generation

For our first two experiments, we performed 117 experiments. Each experiment was performed for one specific camera perspective. The generated data is the same for both. An overview of the parameters used for the generation of the data can be found in Table 1. We use the same data distribution for test and training datasets, but of course, generate them with different seeds. For our third experiment, we generate samples from a uniform distribution across all poses, but for evaluation, we again take the 117 separated test sets. We will now present how we change the individual parameters:

- Identities: The face generation framework is capable of generating images of different identities by sampling from a standard normal distribution and applying the 3D Morphable Model as a statistical model of faces. While generating different identities, it can also generate multiple images of the same identity. We generate 1,000 identities for each experiment. For each identity, we generate 100 images for training and 10 images for testing.
  Apart from images, the framework also generates landmarks associated with each image. It generates 19 landmarks for every image. These landmarks give the position of key points on the face.
  While the facial images and their identities are used for face recognition, the same facial images and their associated landmarks are used for facial landmark detection.
- Expressions: We generated for each of the face images a random expression (following again the 3D Morphable Model and a normal distribution). This introduces yet another layer of complexity in the task which hopefully enables the model to give more comparable performance in real-world scenarios even when it is trained on synthetic data.
- Background: for each image we sample a random background from the DTD dataset [4].
- Yaw: While all the aforementioned parameter distributions remain the same for all the datasets that have been generated, the yaw distribution differs across datasets. As mentioned earlier, yaw is sampled from a Gaussian distribution parametrized via the mean and variance and this parameter controls the viewing direction and therefore visibility in terms of self-occlusion of the face from the side view. While the variance has been set to $20°$ in all cases, the mean differs and that's one of the key factors which cause differences in faces across datasets.
  For the mean of the yaw angle, we take values from $-60°$ to $60°$ which increase by $10°$ for every dataset. So we generate datasets with values of the Yaw mean as $-60°$, $-50°$, $-40°$, ..., $50°$, $60°$. This distribution was used for all testing datasets and the training datasets of experiments 1 and 2. For experiment 3 we used a uniform yaw distribution between $-60°$ and $60°$
- Pitch: Similar to the yaw angle, the pitch also differs across datasets. As mentioned earlier, the pitch is a combination of numerical values that control the visibility of the face from the top and/or bottom view. In the case of pitch, like yaw, the associated parameters are mean and variance. While the variance has been set to $20°$ in all cases, the mean differs and that's one of the key factors which causes differences across datasets.
  The mean of pitch takes values from $-40°$ to $40°$ which increases by $10°$ for every dataset. So we generate datasets with values of the Yaw mean as $-40°$, $-30°$, $-20°$, ..., $30°$, $40°$. This distribution was used for all testing datasets and the training datasets of experiments 1 and 2. For experiment 3 we used a uniform yaw distribution between $-40°$ and $40°$

As the mean of yaw and pitch varies, this variation forms the crux of the need for creating our 117 datasets to mimic different camera positions. Each dataset is a combination of yaw mean and pitch mean to render images from our 1,000 identities that have a specific side view and top/bottom view. Example faces for individual experiments and the average of those distributions are shown in Figure 4

The values we choose for data generation are summarized in Table 1.

3.2. Experiment 1 - Face Recognition

In this task, we perform 117 experiments as mentioned in Section 3.1. In each experiment, we train a model to identify faces as per the identity associated with each face. We create a model in such a way that when an image is presented as input to the trained model, corresponding identity of the image is given as the output. Identities are defined by numbers ranging from 0 to 999.

Each model is trained on a dataset that differs in the context of yaw mean and pitch mean and it gives a training accuracy and a validation accuracy. We use these results (training and validation accuracy) to visualize the performance based on the pose i.e., the distribution of performances given yaw mean and pitch mean combination leading to different accuracy for the face recognition task.

3.3. Experiment 2 - Facial Landmark Detection

In this task also, we perform 117 experiments. We use the same dataset that we have used for the face recognition task. However, here, in each experiment, we train a model to regress 19 landmarks 2D on each face. See figure 3 (b) for the facial landmarks which we are considering.

As for the previous task, each model is trained on a dataset that differs in the context of yaw mean and pitch mean. Each model gives a training accuracy and a validation accuracy at the end of the training and testing phases respectively. We use these results (training and validation accuracy) to visualize the performance based on the pose i.e., the distribution of performances given yaw mean and pitch mean combination leading to different accuracy for the facial landmark detection task.

3.4. Experiment 3 - Uniform Distribution

Whilst in experiments 1 and 2 the distribution of the training and test data was always perfectly matching, we would now like to investigate another interesting scenario: instead of training individual networks per experiment, we train one large network with uniformly distributed data. We would argue, that there should be no bias in the distribution for yaw and pitch this way. Following our hypothesis that the difficulty of the task is dependent on the pose, we would expect to get different results for different poses and results similar to the results in experiment 1. Following the common understanding that bias in performance dominantly arises from bias in the data distribution, we would expect very comparable performance across all different testing pose angles. The test set is the same as for experiment 1 but instead of training 117 individual networks, we train one network following the protocol from experiment one except that the training data is sampled uniformly across poses.

3.5. Experiment 4 - Variance in Results

As we ran each of the 117 sub experiments only once we expect to see slightly different results depending on the random seed during data generation and also during training. We would like to see how high the variation is. Since it is computationally expensive to train all 117 networks for experiments 1 and 2 several times, we perform this analysis only for one sub-experiment. We created 5 different datasets for the frontal setting where yaw mean = 0 and pitch mean = 0, where the datasets differed in seed values. So, the identities generated for all 5 datasets were different. Then, the results of face recognition were analyzed to study the variance in results that arise because of the different identities.

3.6. Assumptions

We have made the following assumptions while conducting the experiments:

- Synthetic Data
  The synthetically generated faces are assumed to map to real-world data. Since our faces lack photorealism it is possible that they miss features that are important for the tasks at hand. We however assume the 3D dependent and low-frequency features to be dominating, especially for the landmark detection task, and those are well represented by our data distribution.

- Bias
  Bias from the pre-trained network (Inception Net which is pre-trained on VGG Faces) can creep into the experiments, for example, the VGG Faces dataset has been trained on frontal faces and we are using datasets where not all faces are frontal. Since we only use a pre-trained network for the face recognition task, but not the landmark detection, we assume this bias to not be the driving factor of our results.

4. Results

The performance evaluation of each experiment, both for training and testing is depicted in 5. The performance evaluation for training the network with uniform data and then testing it for each model is shown in 6

4.1. Experiment 1 and 2

In our first two experiment, we test the face recognition and landmark detection accuracy based on the viewing direction, and the results are visualized in Figure 5. We observe very similar performance of the trained networks for each individual pose distribution on the training data. This suggests that our networks reasonably well converged for each cell. We do already see a slight tendency and slightly worse performance towards larger poses and

Parameter	Value
# identities	1000
# training images per identity	100
# test images per identity	10
# training images	100000
# test images	10000
# landmarks	19
Expressions	On
Background	On
Yaw (camera positions)	-60 to 60 with steps of 10
Pitch (camera positions)	-40 to 40 with steps of 10
Yaw distributions per camera position	Gaussian 0 mean 20 variance
Pitch distributions per camera position	Gaussian 0 mean 20 variance
Illumination	Basel Illumination Prior, empirical
Background	DTD, uniform random
Face Model	BFM 2017 bfm mask, no mouth

Table 1. Data generation summary: overview over all parameters and variables in the data generation process.

especially large positive pitch angles for face recognition. The difference however becomes much more clear on the test dataset which follows the same distribution. Here we see, that the testing accuracy shows a much stronger bias for non-frontal poses. Our experiments lead to a very noisy picture of accuracies which is why we explore the variation for a single cell in experiment 4. Besides the noise, there is however a very clear signal in terms of higher accuracies towards the center and lower accuracies especially towards larger poses. We observe that the exact manifestation of this effect is not identical for the two tasks - e.g. the effect of large pitch seems to affect face recognition more than landmark detection. The effect of bias due to the camera, therefore, seems to be dependent on the task at hand. We also observe that the best result is to be expected slightly of-center, however the precise position is not easy to estimate due to the large noise due to single network training.

4.2. Experiment 3

For experiment 3 we are testing what would happen with a perfectly uniform training distribution. So the training data is unbiased in terms of pose. If the camera position plays a substantial role in the accuracy we can reach per pose we should observe a somewhat similar effect as we observe with the individually trained pose-specific networks from experiment 1. The results are visualized in Figure 6 and we again observe strong differences in the accuracy depending on the pose at hand. The observed results are slightly different from those obtained on the individual pose-dependent training sets and the optimal pose seems to be a camera position slightly from below. As the test data is however identical to experiment 1 we also see, that a pose-specific network performs substantially better on the task of face recognition than a more general one for the good poses. Even under perfectly uniform pose distribution we however observe a strong difference in recognition accuracy suggesting that the difficulty of the task of face recognition indeed depends on the camera position.

4.3. Experiment 4

We observe a high level of noise in our measurements of 5. We were therefore investigating how large the noise is that we would expect. We ran the face recognition experiment 5 times and retrieved a training accuracy of 92.88 on average with a standard deviation of 2.13 and a testing accuracy of 88.96 with a standard deviation of 3.72. This level of noise is observable in the visualizations at hand but we believe that whilst the individual cells are subject to noise we can still retrieve valuable information from the overall picture. If we would increase the compute by a factor of 5 we would get a clearer picture, that would allow to identify the exact best position, but since our data is anyway synthetic we do not think this gives us additional important insights.

4.4. Limitations

The generated faces are far from being photo-realistic. Besides the realism gap arising from the use of 3DMMs, there are also no realistic hair, occluders like glasses, or realistic backgrounds. This domain gap could influence the result of our analysis. In contrast to state-of-the-art GAN-based data generators, we however have full control over the data generation, including full pose variation and illumination variation (compare Section 1.1). In future work, we plan to demonstrate that the effect we show here on synthetic data also arises on more realistic data. With our simplistic model, we first wanted to demonstrate that the effect and influence of the camera pose exists, besides the bias

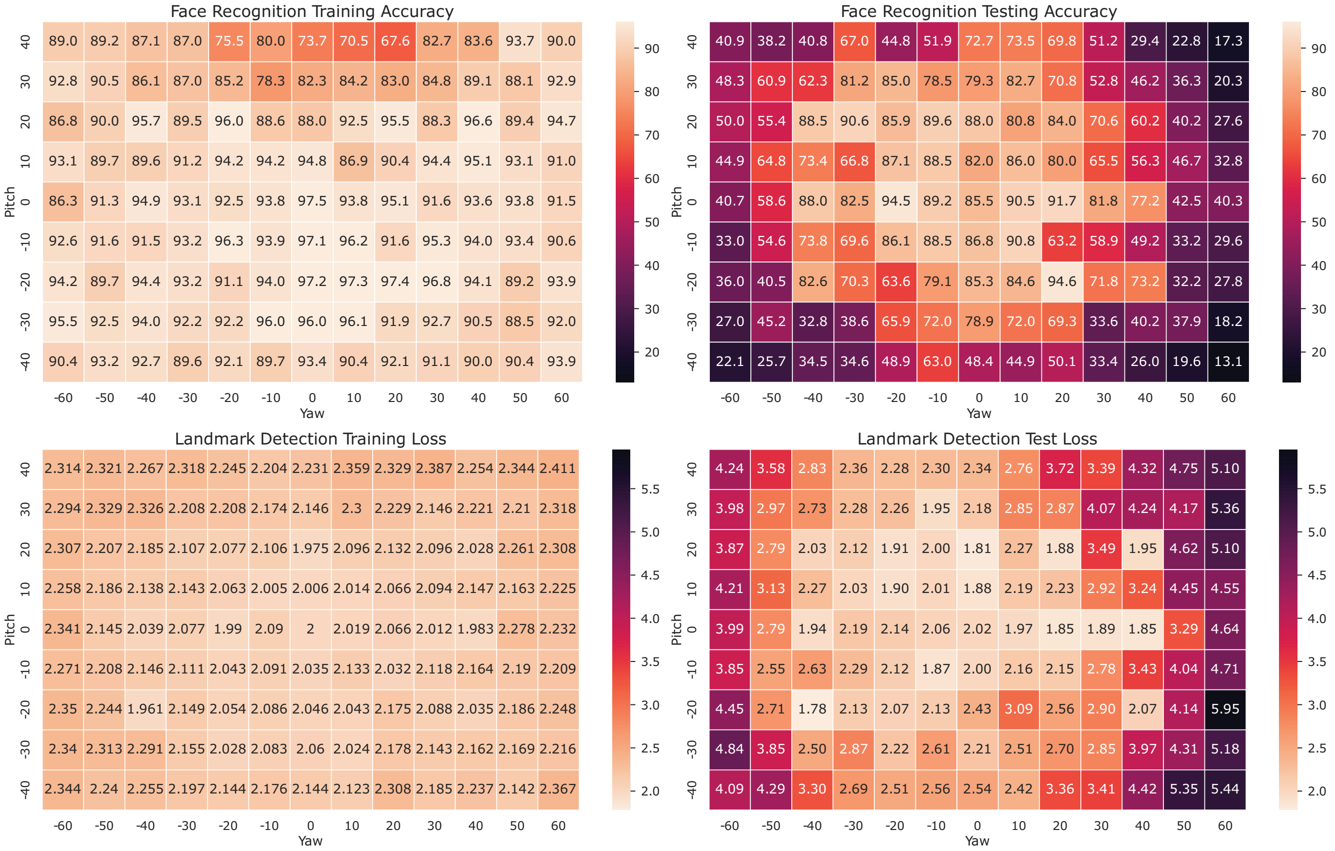

Figure 5. **Performance Visualization of experiments 1 and 2** top left: face recognition-training accuracy, top right: face recognition-testing accuracy, bottom left: landmark detection-training loss, bottom right: landmark detection-test loss.

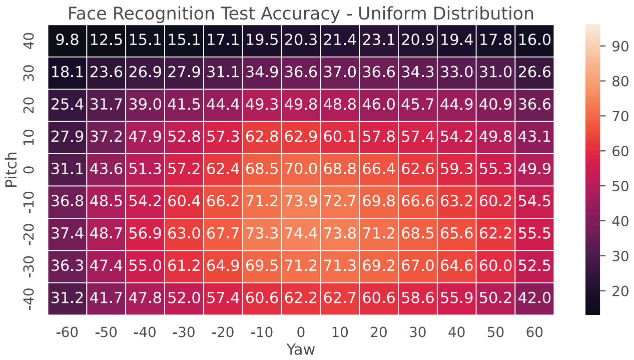

Figure 6. Testing accuracy for face recognition trained on uniformly distributed data set (experiment 3). Training accuracy is 74.82 percent.

in the training data and next we would like to demonstrate that it also arises in real-world settings. Another limitation of our analysis is the high computational demand to perform our evaluation. Whilst the method to generate the data can be easily improved, we need to train networks per pose combination which scales quadratically with the desired resolution of the evaluation grid in yaw and pitch direction. In addition, the results are noisy due to the random nature of the optimization process, ideally, every combination would need to be trained multiple times which we decided to omit (as we can still derive global trends).

5. Conclusion

We demonstrate on ideal synthetic data the effect of task difficulty dependent on the position of the camera. Our experiments suggest that the task difficulty imposed by the camera position is the dominating factor for the reached accuracy and is stronger than the priorly reported pose bias observed in datasets. Our experiments also suggest that the analysis is dependent on the task at hand, which means the analysis has to be done on a per-task basis. For both tasks at hand, we observe a slightly non-frontal pose to be ideal, which is not very surprising as it gives us more depth cues than a purely frontal image and therefore more 3D information. In future work, we will investigate if our findings transfer to photorealistic images and if our method of analysis can be scaled.

6. Acknowledgments

We thank Tobias Koch and Felix Lugauer-Breunig for interesting discussions and feedback. This project was funded via the FAU Emerging Talent Initiative (ETI). The authors gratefully acknowledge the HPC resources provided by the Erlangen National High Performance Computing Center (NHR) at FAU.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[2] Vicki Bruce, Tim Valentine, and Alan Baddeley. The basis of the 3/4 view advantage in face recognition. *Applied cognitive psychology*, 1(2):109–120, 1987.

[3] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020.

[6] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE, 2018.

[7] Marcel Grimmer, Christian Rathgeb, and Christoph Busch. Pose impact estimation on face recognition using 3d-aware synthetic data with application to quality assessment. *arXiv preprint arXiv:2303.00491*, 2023.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[11] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2093–2102, 2018.

[12] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[13] Adam Kortylewski, Andreas Schneider, Thomas Gerig, Bernhard Egger, Andreas Morel-Forster, and Thomas Vetter. Training deep face recognition systems with synthetic data. *arXiv preprint arXiv:1802.05891*, 2018.

[14] Frances L Krouse. Effects of pose, pose change, and delay on face recognition performance. *Journal of Applied Psychology*, 66(5):651, 1981.

[15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. 06 2016.

[17] Philippe G Schyns and Heinrich H Bulthoff. Viewpoint dependence and face recognition. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* , pages 789–793. Routledge, 2019.

[18] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gérard Medioni. Gan-control: Explicitly controllable gans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[20] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1283–1292, 2017.

[21] Nikolaus F Troje and Heinrich H Bülthoff. Face recognition under varying poses: The role of texture and shape. *Vision research*, 36(12):1761–1771, 1996.

[22] Seyma Yucer, Furkan Tektas, Noura Al Moubayed, and Toby P Breckon. Racial bias within face recognition: A survey. *arXiv preprint arXiv:2305.00817*, 2023.