To Frontalize or Not To Frontalize: A Study of Face Pre-Processing Techniques and Their Impact on Recognition

Sandipan Banerjee*, Joel Brogan*, Janez Križaj2, Aparna Bharati1, Brandon Richard Webster1, Vitomir Štruc2, Patrick J. Flynn1 and Walter J. Scheirer1
1 Dept. of Computer Science & Engineering, University of Notre Dame, USA
2 Faculty of Electrical Engineering, University of Ljubljana, Slovenia

{\text{sbanerj1, jbrogan4, abharati, bricharl, flynn, wscheire}@nd.edu}
{\text{janez.krizaj, vitomir.struc}@fe.uni-lj.si}

Abstract

Face recognition performance has improved remarkably in the last decade. Much of this success can be attributed to the development of deep learning techniques such as convolutional neural networks (CNNs). While CNNs have pushed the state-of-the-art forward, their training process requires a large amount of clean and correctly labelled training data. If a CNN is intended to tolerate facial pose, then we face an important question: should this training data be diverse in its pose distribution, or should face images be normalized to a single pose in a pre-processing step? To address this question, we evaluate a number of popular facial landmarking and pose correction algorithms to understand their effect on facial recognition performance. Additionally, we introduce a new, automatic, single-image frontalization scheme that exceeds the performance of current algorithms. CNNs trained using sets of different pre-processing methods are used to extract features from the Point and Shoot Challenge (PaSC) and CMU Multi-PIE datasets. We assert that the subsequent verification and recognition performance serves to quantify the effectiveness of each pose correction scheme.

1. Introduction

The advent of deep learning [1] methods such as convolutional neural networks (CNNs) has allowed face recognition performance on hard datasets to improve significantly. For instance, Google FaceNet [2], a CNN based method, achieved over 99% verification accuracy on the LFW dataset [3], which was once considered to be extremely challenging due to its unconstrained nature. Because CNNs possess the ability to automatically learn complex representations of face data, they systematically outperform older methods based on hand-crafted features. Since these representations are learned from the data itself, it is often assumed that we must provide CNNs clean, pre-processed data for training. Accordingly, complex frontalization steps are thought to be integral to improving CNN performance [4]. However, with the use of a pose correction method comes many questions: How extreme of a pose can the frontalization method capture? Should the method enforce facial symmetry? Does training CNNs with frontalized images yield better results, or can CNNs automatically learn robust representations invariant of facial pose on their own? Currently, a thorough evaluation of the effects of frontalization on face recognition is lacking. To fill this gap, we conducted an extensive comparative study of different facial pre-processing techniques.

For this study, we used the CASIA-WebFace [5] dataset for CNN training. Two popular frontalization techniques were chosen for our training and testing evaluation: the well-established method proposed by Hassner et al. (H) [7], and our own newly proposed method. Furthermore, to evalu-
evaluate the effect of facial landmarking on the frontalization process, we used three landmarking techniques: Zhu and Ramanan (ZR) [6], Kazemi and Sullivan (KS) [8], and a novel technique - a Cascade Mixture of Regressors (CMR). Different frontalization results using various combinations of these methods can be seen in Fig. 1.

We used the popular VGG-FACE CNN [9] as our base architecture for training networks using different pre-processing strategies. The PaSC video dataset [10] was then used for testing. We extracted face representations from individual video frames in PaSC using a network trained with a particular pre-processing strategy. These features were used for verification and recognition purposes by applying a cosine similarity score-based face matching procedure. Finally, we compared our results to a set of baseline networks trained using non-frontalization methods. The effect of each data augmentation is manifested in the performance of each subsequent CNN model.

In summary, the contributions of this paper are:

- The evaluation of popular facial landmarking and frontalization methods to quantify their effect on video-based face recognition tasks using a CNN.
- A determination of whether symmetric or asymmetric frontal reconstruction works best for face recognition.
- An investigation of frontalization failure rates as a function of pose using CMU Multi-PIE [11].
- A new, effective facial landmarking and frontalization technique for comparison with the other methods.
- Experiments that show our landmarker and frontalizer to be separately compatible with other frontalization and landmarking methods respectively.
- Experiments that show that our frontalization method outperforms other methods in the study, when both the training and testing data are frontalized.

2. Related Work

Previous work relevant to this subject can be categorized into three broad groups as listed below.

**Facial landmarking:** Facial landmarks are used in frontalization to determine transforms between a facial image and template. Over the past decade, an array of landmarking techniques have been developed that rely on handcrafted features [8]. Recently, deep learning has been used for landmark training and regression [12]. Current algorithms provide landmark sets of a size between 7 and 194 points. Of late, landmarkers have begun to conform to a 68-point standard to improve comparative analysis between algorithms, and across different landmarking challenges and datasets [6, 8, 13].

**Face frontalization:** Once facial landmarks are detected on a non-frontal face, frontalization can be performed using one of the two main approaches. The first approach utilizes 3D models for each face in the gallery, either inferred statistically [14], collected at acquisition time [15], or generic [7]. Once the image is mapped to a 3D model, matching can be performed by either reposing the gallery image to match the pose of the query image or the query image can be frontalized [16]. These methods have been utilized in breakthrough recognition algorithms [4]. The second approach uses statistical models to infer a frontal view of the face by minimizing off-pose faces to their lowest rank reconstruction [17]. Additionally, methods have been explored for inferring frontal faces using deep learning [18].

**Face recognition:** In its infancy, face recognition research used handcrafted features for representing faces [19]. More recently, state-of-the-art deep CNN methods for face recognition have achieved near-perfect recognition scores on the once-challenging LFW dataset [3] using learned representations. While some of these methods concentrate on creating novel network architectures [9], others focus on feeding a large pool of data to the network training stage [4, 2]. Researchers have now shifted their attention to the more challenging problem of face recognition from videos. The Youtube Faces (YTF) dataset [20], IJB-A [21] and PaSC [10] exemplify both unconstrained and controlled video settings. Researchers have used multi-pose based CNN models [22, 23] or have exploited face frontalization as a data augmentation step [24] for recognizing faces from these challenging video datasets.

3. Description of Chosen Landmarking & Frontalization Methods

Here we present brief descriptions of the facial landmarking and frontalization techniques used in this paper.

3.1. Landmarking

**Zhu and Ramanan (ZR) [6]:** The ZR method allows for simultaneous face detection, landmarking, and pose detection, accommodating up to 175 degrees of facial yaw. ZR uses a mixture of trees approach, similar to that of phylogenetic inference. The algorithm proposed in [25] is used to optimize the tree structure with maximum likelihood calculations based on training priors. Due to the algorithm performing localization and landmarking concurrently, it is relatively slow.

**Kazemi and Sullivan (KS) [8]:** KS uses a cascade of multiple regressors to estimate landmark points on the face using only a small, sparse subset of pixel intensities from the image. This unique sub-sampling renders it extremely fast, while maintaining a high level of accuracy. This land-
marker is popular due to its ease of use and availability — it is implemented in the widely used Dlib library [26].

Cascade Mixture of Regressors (CMR): We introduce the CMR method as a fast performing alternative for facial landmarking that works well on lower-resolution images in datasets such as CASIA-Webface and PaSC. Similar to the Supervised Descent Method [27], this method initializes its set of landmarks in a defined initial formation around the detected face. Then a regression function in the form of a Gaussian mixture of regressors is applied to local features extracted from the initial landmark locations. The landmark positions are fine-tuned iteratively using a cascade of mixture regressors similar to [28].

3.2. Frontalization

Hasner et al. (H) [7]: This method allows 2D face images to be frontalized without any prior 3D knowledge. We chose to analyze this method due to its prominence in the facial biometrics community, and because an open source implementation of the algorithm exists. Using a set of reference 3D facial landmark points determined by a 3D template, the 2D facial landmarks detected in an input image are projected into the 3D space. A 3D camera homography is then estimated between them. Back-projection is subsequently applied to map pixel intensities from the original face onto the canonical, frontal template. Optional soft symmetry can be applied by replacing areas of the face that are self-occluded with corresponding patches from the other side. Due to the global projection of this method, incorrect landmarking can stretch and distort the frontalized face, causing loss of high-frequency features used for matching.

4. Proposed Frontalization Method

In this section, we present our proposed frontalization procedure, which is capable of synthesizing a frontalized face image from a single input image with arbitrary facial orientation without requiring a subject-specific 3D model.

4.1. Face Detection, Landmarking & Model Fitting

Our proposed frontalization procedure starts (see Fig. 2 (a)) by detecting the facial region in the input image \( I_0 \) using the Viola-Jones face detector [29]. Using the CMR method, we detect 68 facial landmark points, i.e., \( p_0 = [x_1, y_1, \ldots, x_{68}, y_{68}]^\top \in \mathbb{R}^{2 \times 68} \). The landmarks can be used to determine the pose and orientation of the processed face. We crop the facial area, \( I_c \), from the input image based on the detected landmarks and use it as the basis for frontalization.

To transform the face in the input image to a frontal pose, we require a depth estimate for each of the pixels in the cropped facial area. To this end, we use a generic 3D face model and fit it to the cropped image \( I_c \). Our 3D model is a frontal depth image \( I_r \) from the FRGC dataset [19] manually annotated with the same 68 landmarks as detected by the CMR procedure. We fit the 3D model to the cropped image through a piece-wise warping procedure guided by the Delaunay triangulation of the annotated landmarks. Since the annotated landmarks reside in a 3D space, i.e., \( p_r = [x_1, y_1, z_1, \ldots, x_{68}, y_{68}, z_{68}]^\top \in \mathbb{R}^{3 \times 68} \), we use only the 2D coordinates in the XY-plane to compute the triangulation. The fitting procedure then aligns the generic 3D model with the shape of the cropped image and provides the depth information needed for the 3D transformation of the input face to a frontal pose (see Fig. 2 (b)). The depth information generated by the warping procedure represents only a rough estimate of the true values, but, as we show later, is sufficient to produce visually convincing frontalization results.

4.2. 3D Transformation & Texture Mapping

After the fitting process, we use the landmarks \( p_a \in \mathbb{R}^{3 \times 68} \) corresponding to the aligned 3D model \( I_a \) and the landmarks \( p_c \in \mathbb{R}^{3 \times 68} \) of the generic 3D face model to estimate a 3D transformation, \( T \in \mathbb{R}^{4 \times 4} \), that maps the fitted model \( I_a \) back to frontal pose (Fig. 2 (c)). We use Horn’s quaternion based method [30] to calculate the necessary scaling, rotation and translation to align the 3D points in \( p_a \) and \( p_c \) and construct the transformation matrix \( T \). The results of the presented procedure are shown in Fig. 2 (d). Here, images in the upper row illustrate the transformation of the 3D models in accordance with \( T \), while the lower row depicts the corresponding texture mapping. The mapped texture image \( I_t \) represents an initial frontal view of the input face, but is distorted in some areas. We correct for these distortions with the postprocessing steps described in the next section.

4.3. Image Correction & Postprocessing

Similar to the method of [7], our approach utilizes a generic 3D face model to generate frontalized face images. Unlike in [7], we adapt our model in accordance with the shape of the input face to ensure a better fit. Triangulation is performed on the input face landmark coordinates. Each triangle is then mapped back to the generic 3D face model,
and an affine transform is calculated per-triangle. Because the piecewise alignment is performed with a warping procedure, minor distortions are introduced into the shape of the aligned 3D model, which lead to artifacts in the mapped texture image $I_f$. Additional artifacts are also introduced by the interpolation procedure needed to compensate for the obscured or occluded areas in the input images caused by in-plane rotations and self-occlusions.

We correct for the outlined issues by analyzing the frontalized 3D model $I_f$. Since Eq. (1) defines a mapping from $I_a$ to $I_f$, the frontalized 3D model $I_f$ is not necessarily defined over a rectangular grid, but in general represents a point cloud with areas of different point density. We identify obscured pixels in $I_a$ based on point densities. If the density for a given pixel falls below a particular threshold, we mirror the corresponding pixel from the other side of the face to form a more symmetric face.

The effect of the presented image correction procedure is illustrated in Fig. 2 (e). The image, marked as $I_m$, contains white patches that were identified as being occluded in $I_a$, while $I_o$ represents the corrected image with pixels mirrored from one side of the face to the other (examine the difference in the appearance of the nostrils between $I_t$ and $I_n$). In the last step of our frontalization procedure we map the image $I_m$ to a predefined mean shape. This mapping generates the final frontalized output $I_1$ of our procedure and is shown in the last image of Fig. 2 (e).

5. Face Recognition Pipeline

In this section, we provide details about our face recognition pipeline.

5.1. Training Data: CASIA-WebFace

The CASIA-WebFace [5] dataset contains 494,414 well-labeled face images of 10,575 subjects, with 46 face images per subject on average. The dataset contains face images of varying gender, age, ethnicity and poses, and was originally released for training CNNs. In comparison, MegaFace [31] and VGG-FACE [9] contain over a million face images, but have significantly more labeling errors. For this reason, coupled with what was feasible to process with available GPU hardware, we ultimately chose a reduced subset of CASIA-WebFace, containing 303,481 face images of 7,577 subjects, as our training dataset.

5.2. Pre-processing Methods

The pre-processing schemes used in our experiments were comprised of different combinations of landmarkers and frontalizers described in Secs. 3 and 4: 1) ZR [6] & H [7], 2) KS [8] & H [7], 3) CMR & our frontalization method, 4) CMR & H [7], 5) ZR [6] & our frontalization method, and 6) KS [8] & our frontalization method.

In addition, we compared these methods to three baseline approaches: 1) Training VGG-FACE with only 2D aligned CASIA-WebFace images, rotated using eye-centers, i.e., no frontalization (Figure 1.b). 2) Training VGG-FACE with original CASIA-WebFace images, i.e., no pre-processing. 3) A snapshot of the original VGG-FACE model, pre-trained on 2D aligned face images from the VGG-FACE dataset [9], as a comparison against a prevalent CNN model.

5.3. CNN architecture: VGG-FACE

The CNN architecture in this work is similar to the one described by Parkhi et al. [9], a variant of the 16 layer model proposed in [32]. The architecture is comprised of multiple linear convolution layers, with Rectified Linear Unit (ReLU) and max pooling layers between each. These are followed by three fully connected (FC) layers with a filter size that is the same as the input image (224×224). The first two FC layers (denoted as fc6 and fc7) are 4,096 dimensional, while the size of the final layer (fc8) depends on the number of classes in the training data.
This particular architecture was chosen because it generates verification results comparable to Google FaceNet [2] on LFW [3] while requiring a fraction of its training data. Additionally, the model performs reasonably well on the YTF [20] video dataset. Lastly, a snapshot of this model pre-trained on 2 million face images is present in the Caffe [33] model zoo. We used this pre-trained model to fine-tune connection weights in our training experiments for faster convergence.

5.4. Testing Datasets

For completeness, we perform two types of frontalization tests to gain a more holistic understanding of the behavior of different frontalizer schemes. The first set of tests, which analyze the performance impact of different frontalization methods on facial recognition, utilizes the PaSC Dataset [10]. The second set of tests are designed to analyze the yield rates and failure modes of frontalizers for different pose conditions. For these tests, we utilize the CMU MultiPIE dataset [11]

PaSC - The PaSC dataset [10] is a collection of videos acquired at the University of Notre Dame over seven weeks in the Spring semester of 2011. The human participants in each clip performed different pre-determined actions each week. The actions were captured using handheld and stationary cameras simultaneously. The dataset contains 1,401 videos from handheld cameras and 1,401 videos from a stationary camera. A small training set of 280 videos is also available with the dataset.

While both YTF [20] and IJB-A [21] are well-established datasets, they are collections of video data from the Internet. On the other hand, PaSC consists of video sequences physically collected specifically for face recognition tasks. This type of controlled acquisition is ideal for our video-to-video matching-based evaluation.

MultiPIE - To evaluate the success rate of each landmark and frontalizer combination at specific facial pose angles (yaw), we used the CMU Multi-PIE face database [11]. This dataset contains more than 750,000 images of 337 different people. We utilize the multipose partition of the dataset, containing 101,100 faces imaged under 15 view points with differing yaw angles and 19 illumination conditions, with a variety of facial expressions. For pose consistency, we exclude the set of view points that also induce pitch variation.

5.5. Feature Extraction and Scoring

We used networks trained on data pre-processed with each of the combinations mentioned above as feature extractors for PaSC video frames. Before the feature extraction step, the face region from each frame was extracted using the bounding box provided with the dataset. Bad detections were filtered by calculating the average local track trajectory coordinates to roughly estimate the locations of neighboring detections, and removing detections with coordinates outside a $2.5\sigma$ (standard deviation) distance range from their estimated location.

After pose correction, a 4,096 dimensional feature vector was extracted at the $fc7$ layer for every face image using each CNN model. Once feature vectors for all frames were collected, the accumulated feature-wise means at each dimension were calculated to generate a single representative vector for that video. This accumulated vector can be represented as $[f_1, f_2, f_3, \ldots, f_{4096}]$, such that

$$f_k = \frac{1}{N} \sum_{i=1}^{N} (v_k)_i$$

where $(v_k)_i$ is the $k$-th feature in frame $i$ of the video and $N$ is the total number of frames in that video.

Cosine similarity was then used to compute match scores between different accumulated feature vectors from two different videos, as shown below:

$$S(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2}$$

where $S(u,v)$ is the similarity score between the two vectors $u = [f_1, f_2, f_3, \ldots, f_{4096}]$ and $v = [f_1, f_2, f_3, \ldots, f_{4096}]$. If $u$ and $v$ are perfectly alike (0 angle), then $S(u,v)$ is 1. Ideally, two videos (vectors) of the same subject should generate a cosine similarity score of approximately 1, i.e., a true match. These similarity scores were used for calculating the verification and identification accuracy rates of each CNN.

6. Experiments & Results

In this section we present details about our experiments and the subsequent results.

6.1. Methodology

To analyze the effect that facial frontalization has on recognition performance, we performed two experiments: 1) training VGG-FACE with face images pre-processed using different strategies, then analyzing its performance on only the 2D aligned PaSC dataset, i.e., frontalization at training time, and 2) frontalizing the PaSC dataset as well i.e., congruent frontalization for both training and testing.

For each frontalization method, we kept two pre-processed versions of the same face: one without any symmetry (asymmetric), such as the left hand side of Figure 1 (c), and the other with symmetry, where one vertical half is used for both sides of the face, as in the right hand side of Figure 1 (c). For the symmetric versions, the half to replicate was chosen automatically based on the quality of the facial landmark points.
The intricacy of face frontalization methods causes them to have increased modes of failure, with decreased operational ranges, as compared to simpler 2D alignment methods. For most frontalization strategies, many samples in CASIA-WebFace fail in the landmarking or frontalizing step due to extreme scale, pose (> 45° yaw), or occlusion. Hence each pre-processing method yields a unique subset of frontalizable images well below the total original number. The yield of frontalized faces varies for each combination, which can be seen in Table 1. Because this work aims to analyze the performance of end-to-end pipelines for facial recognition using different frontalization techniques, it was decided to directly compare results on these different subsets of PaSC, instead of only utilizing test examples falling within the union of all subsets.

To understand the failure modes of each frontalization scheme, we further frontalized face images from the multi-view partition of the CMU Multi-PIE dataset [11]. All six frontalization techniques (ZR & H, KS & H, CMR & our method, CMR & H, ZR & our method and KS & our method) were tested for each pose in the dataset, including differing expressions and illumination. The pose angles tested were binned into subsets of 0°, 15°, 30°, 40°, 60°, 70° and 90°, along with respective negative angles. Angle bins containing images taken with cameras with yaw and pitch angles were excluded from this experiment. Failures from landmarking or from frontalization itself were not differentiated. The results can be seen in Figure 3.

In general, all methods often see high failure rates on facial pose angles beyond 40°. Methods using CMR for landmarking perform best in the 0 - 40° range. Our method causes slightly more failures than H [7] within a +/−40° range, but has equal performance on more extreme poses. KS [8] provides superior performance on extreme poses (ZR’s [6] profile landmarker is not used in this study, as we deliberately chose not to include pose estimation).

We trained a VGG-FACE model separately for each subset of training data pre-processed with a given method. For each method, we randomly partition 90% of CASIA-WebFace for training, with 10% for validation. A single NVIDIA Titan X GPU was used to run all of our training experiments using the Caffe [33] deep learning package. Network weights were initialized from a snapshot of VGG-FACE pre-trained on 2 million face images. We used Stochastic Gradient Descent [34] for CNN training. The set of hyperparameters for this method was selected using HyperOpt [35] and the same set was repeated across the different experiments to maintain consistency. The base learning rate was set to 0.01, which was multiplied by a factor of 0.1 (gamma) following a stepwise learning policy, with step size set to 50,000 training iterations. The training batch size was set to 64, with image resolution of 224×224. The snapshot at the 50th epoch was then used for feature extraction in the testing phase.

6.2. Results of Recognition Experiments

In the first set of recognition experiments, we used a uniform subset of 2D aligned face images extracted from PaSC for testing. For each experiment, we computed verification performance with a ROC curve, as well as the rank-based recognition performance, i.e., identification using a CMC curve. Verification and identification performance measures are pertinent in analyzing the behavior of each frontaliza-
Table 1: Yield of each pre-processing method with best replication mode listed (asym - asymmetric, sym - symmetric). “OFM” represents our frontalization method.

| Pre-processing method | ZR & H | KS & H | CMR & OFM | CMR & H | ZR & OFM | KS & OFM | 2D alignment (not frontalized) |
|-----------------------|--------|--------|-----------|---------|----------|---------|-------------------------------|
| CASIA-WebFace images  | 261,951 | 294,020 | 252,222 (sym) | 252,294 (sym) | 254,381 (asym) | 266,269 (asym) | 268,455 |
| Yield (%)             | 86.31  | 96.88  | 83.11     | 83.13   | 83.82    | 87.74   | 88.45                         |

Figure 5: Verification performance on 2D aligned handheld PaSC videos. “OFM” represents our frontalization method. The 2D aligned curve gave the best performance, while training with the non pre-processed images actually hampered the CNN’s face representation capability (dotted curve).

Figure 6: Recognition performance on frontalized handheld PaSC videos. Pre-processing both the training and testing data with KS [8] & our frontalization method (OFM) outperforms all other methods, and exceeds the best performance reported in Fig. 4.

To evaluate the effect of frontalization at the time of testing, we test each network trained on frontalized data with its respective PaSC subset, frontalized using the same scheme. The identification and verification performance of these models can be found in Figures 6 and 7 respectively. We find that pre-trained VGG always performs consistently (25 - 30% rank-1 accuracy) for any mode of training frontalization, while and the network with no pre-processing always performs poorly (rank-1 accuracy < 5%). That is why they are not presented in these plots.

Pre-processing both CASIA-WebFace and PaSC using the KS [8] landmarker coupled with our frontalization method produces the best results with VGG-FACE. The rank-1 accuracy improves overall compared to the 2D-alignment pre-processing strategy in Figure 4. Our frontalization method outperforms H [7] in almost all cases, i.e., using different landmarkers. We attribute this to the local adaptation of our 3D model described in Section 4.3, in contrast to using a general model as in H [7], which can stretch and distort faces (see Section 3.2). This ensures a better fit to the input face and the frontalized face image looks more...
natural and preserves higher-frequency features as a result, which can be seen in Figures 1.d and 1.h.

Surprisingly, training the network with 2D-aligned face images negatively affects recognition performance regardless of the pre-processing method used to frontalize PaSC, which suggests that performing frontalization at testing time does not benefit performance on pre-trained networks. Instead, training and testing must be pre-processed under consistent methods to realize any performance benefit.

Another recurring trend that can be noticed in Figures 4 and 6 is that recognition performance is slightly improved when the face images are reconstructed asymmetrically rather than symmetrically. While symmetrically reconstructing faces can provide a more visually appealing result, important data still present in the occluded side of an off-pose face can be destroyed by such operations. By superimposing portions of the non-occluded face regions to fill in gaps on the occluded side, artifacts are inevitably introduced onto the reconstructed face. We suspect these artifacts to be detrimental to the feature learning of a CNN, and consequently its recognition performance suffers.

7. Discussion

Several conclusions can be drawn from our experiments and used to moderate future face recognition experiments:

1. Frontalization is a complex pre-processing step, meaning it can come at a cost. Due to the large number of failure modes frontalization introduces, there can be significant loss of data, i.e., lower yield, specifically with images containing extreme pose or occlusion. Additionally, frontalization can prove to be computationally expensive, meaning the performance benefit frontalization can provide must be weighed against the needed increase in computational resources.

2. Frontalization at training time does not improve performance unless the testing data is processed in a manner consistent with the training set. Conversely, frontalization of testing data provides performance increase on pre-trained CNNs only if the network is trained using data pre-processed in a consistent manner to the testing set.

3. While symmetrically reconstructed frontalized faces may yield more visually appealing results, asymmetrical frontalization provides superior performance for face recognition.

4. Our proposed method, which dynamically adapts local areas of the 3D reference model to the given input face, provides better performance improvements than that of Hassner et al. [7] for PaSC video recognition.

5. Training a CNN with millions of face images makes it relatively agnostic (in terms of recognition performance) to the pre-processing method used on testing data, e.g., pre-trained VGG-FACE performs consistently across different experiments.

From these observations, we can conclude that the usefulness of frontalization to pre-process test set faces can be dependent on the facial recognition system used. Depending on how the recognition system in question was trained, including a frontalization step may cause domain mismatches, as noted in Section 6.2, which can in fact degrade recognition performance. Therefore, face frontalization should be taken with a grain of salt, as it may not always provide superior results.

We plan to extend this work further by checking the cross-compatibility of each landmarking and frontalization method with the others by pre-processing the training data with one method and pre-processing the testing data with a different method. We wish to see how flexible models trained with different frontalization schemes are towards testing data that has been frontalized slightly differently. Additionally, we want to perform a human rater study to compare verification performance on frontalized face images between a human and a CNN. This experiment should help us identify how the visual quality of a frontalization method correlates to recognition performance, both in humans and CNNs.

Acknowledgements: Hardware support was generously provided by the NVIDIA Corporation.
References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[2] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proc. IEEE CVPR*, 2015.
[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Technical Report 07-49, UMass, Amherst, Tech. Rep., 2007.
[4] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proc. IEEE CVPR*, 2014.
[5] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” *arXiv:1411.7923*.
[6] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *Proc. IEEE CVPR*, 2012.
[7] T. Hassner, S. Harel, E. Paz, and R. Enbar, “Effective face frontalization in unconstrained images,” in *Proc. IEEE CVPR*, 2015.
[8] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Proc. IEEE CVPR*, 2014.
[9] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *Proc. BMVC*, 2015.
[10] W. Scheirer et al., “Report on the btas 2016 video person recognition evaluation,” in *Proc. IEEE BTAS*, 2016.
[11] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, “Multi-pie,” *Image Vision Computing.*, vol. 28, no. 5, pp. 807–813, 2010.
[12] Y. Wu and T. Hassner, “Facial landmark detection with tweaked convolutional neural networks,” *arXiv:1511.04031*.
[13] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, “The first facial landmark tracking-in-the-wild challenge: Benchmark and results,” in *Proc. IEEE ICCV Workshops*, 2015.
[14] T. Hassner, “Viewing real-world faces in 3d,” in *Proc. IEEE CVPR*, 2013.
[15] V. Blanz and T. Vetter, “Face recognition based on fitting a 3d morphable model,” *IEEE TPAMI*, vol. 25, no. 9, pp. 1063–1074, 2003.
[16] X. Zhang and Y. Gao, “Face recognition across pose: A review,” *Pattern Recognition*, vol. 42, no. 11, pp. 2876–2896, 2009.
[17] C. Sagonas, Y. Panagakis, S. Zafeiriou, and M. Pantic, “Robust statistical face frontalization,” in *Proc. IEEE ICCV*, 2015, pp. 3871–3879.
[18] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim, “Rotating your face using multi-task deep neural network,” in *Proc. IEEE CVPR*, 2015.
[19] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, “Overview of the face recognition grand challenge,” in *Proc. IEEE CVPR*, 2005.
[20] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *Proc. IEEE CVPR*, 2011.
[21] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain, “Pushing the frontiers of unconstrained face detection and recognition: iarpa janus benchmark a,” in *Proc. IEEE CVPR*, 2015.
[22] I. Masi, S. Rawls, G. Medioni, and P. Natarajan, “Pose-Aware Face Recognition in the Wild,” in *Proc. IEEE CVPR*, 2016.
[23] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajan, R. Nevatia, and G. Medioni, “Face recognition using deep multi-pose representations,” in *Proc. IEEE WACV*, 2016.
[24] I. Masi, A. T. Trân, T. Hassner, J. T. Lekust, and G. Medioni, “Do we really need to collect millions of faces for effective face recognition?” in *Proc. ECCV*, 2016.
[25] S. Kirshner, P. Smyth, and A. W. Robertson, “Conditional chow-liu tree structures for modeling discrete-valued vector time series,” in *Proc. UAI*, 2004.
[26] D. E. King, “Dlib-ml: A machine learning toolkit,” *JMLR*, vol. 10, no. Jul, pp. 1755–1758, 2009.
[27] X. Xiong and F. De la Torre, “Supervised descent method and its applications to face alignment,” in *Proc. IEEE CVPR*, 2013.
[28] T. Oncel, S. Tambe, and T. K. Marks, “Robust face alignment using a mixture of invariant experts,” in *Proc. ECCV*, 2016.
[29] P. Viola and M. J. Jones, “Robust real-time face detection,” *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.
[30] B. K. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.
[31] I. Kemelmacher-Shlizerman, S. Seitz, D. Miller, and E. Brossard, “The megaface benchmark: 1 million faces for recognition at scale,” in *Proc. IEEE CVPR*, 2016.
[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Proc. ICLR*, 2015.
[33] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proc. ACM MM*, 2014.
[34] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proc. COMPSTAT*, 2010.
[35] J. Bergstra, D. Yamins, and D. D. Cox, “Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms,” in *Proc. SciPy*, 2013.
(a) Recognition performance on 2D aligned stationary PaSC videos.
(b) Verification performance on 2D aligned tripod mounted PaSC videos.
(c) Recognition performance on frontalized stationary PaSC videos.
(d) Verification performance on frontalized tripod mounted PaSC videos.
Figure 8: Randomly selected frames from - (a) 2D aligned tripod mounted, (b) 2D aligned handheld, (c) frontalized tripod mounted, (d), frontalized handheld PaSC videos. Columns (1) - (4) represent different subjects in the PaSC dataset. Note the difference in resolution, illumination, occlusion and facial pose in the different frames.