Eurographics Workshop on Intelligent Cinematography and Editing (2022)
R. Ronfard and H.-Y. Wu (Editors)

Framework to Computationally Analyze Kathakali Videos

Pratikkumar Bulani[1], Jayachandran S[2], Sarath Sivaprasad[1,3] and Vineet Gandhi[1]

[1]CVIT & KCIS, IIIT Hyderabad, India
[2]Center for Exact Humanities, IIIT Hyderabad, India
[3]TCS Research, Pune, India

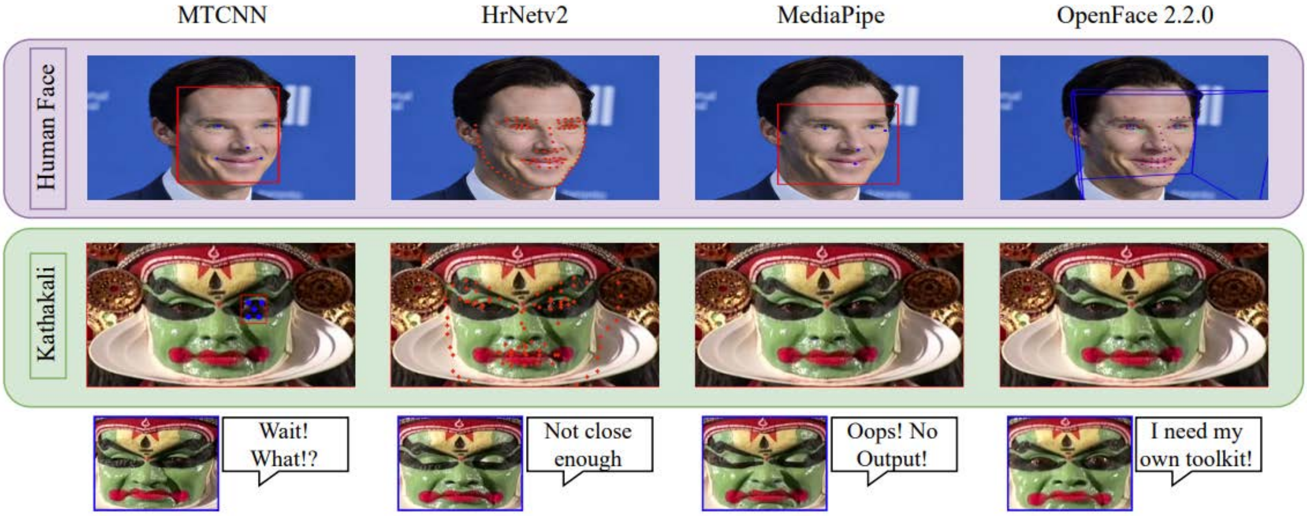


Figure 1: *Existing face analysis pipelines do not work on* Kathakali *faces. Therefore our work proposes a novel toolbox for analysis of* Kathakali *videos.*

Abstract

Kathakali *is one of the major forms of Classical Indian Dance. The dance form is distinguished by the elaborately colourful makeup, costumes and face masks. In this work, we present (a) a framework to analyze the facial expressions of the actors and (b) novel visualization techniques for the same. Due to extensive makeup, costumes and masks, the general face analysis techniques fail on* Kathakali *videos. We present a dataset with manually annotated* Kathakali *sequences for four downstream tasks, i.e. face detection, background subtraction, landmark detection and face segmentation. We rely on transfer learning and fine-tune deep learning models and present qualitative and quantitative results for these tasks. Finally, we present a novel application of style-transfer of* Kathakali *video onto a cartoonized face. The comprehensive framework presented in the paper paves the way for better understanding, analysis, pedagogy and visualization of* Kathakali *videos.*

Demo output: https://github.com/pratikk-bulani/kathakali_analysis

CCS Concepts

• ***Applied computing*** → *Performing arts;* • ***Human-centered computing*** → *Visualization;*

1. Introduction

Kathakali conveys stories through performance art and is a major form of classical Indian Dance. The name derives from the words *Katha* means story or a traditional tale and *Kali* meaning performance or a play. The performer does not use verbal cues in the storytelling process. The actors use *Mudras*, the hand gestures to convey the dialogue of their individual character. The internal state of their character is portrayed using facial expressions or the *Bhava*.

In this work, we focus on detecting faces and analyzing the *Bhavas* or the facial expressions of the *Kathakali* actors. A customized computational *Bhava* understanding pipeline will help improve understanding of *Kathakali* videos and widen its coverage. It

DOI: 10.2312/wiced.20221049

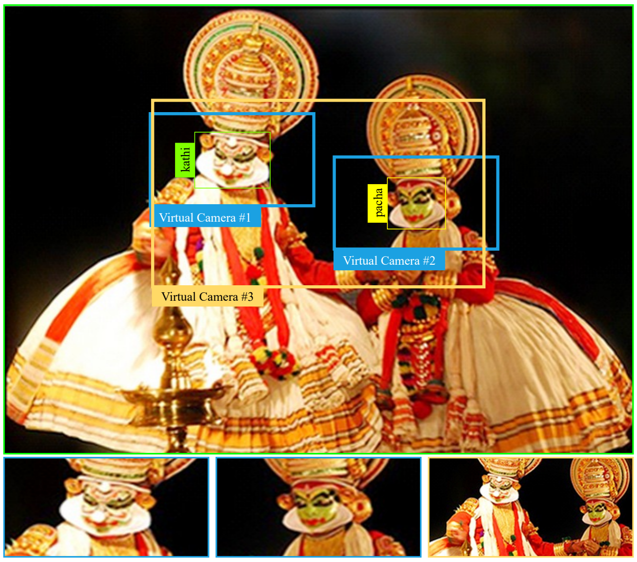


Figure 2: *A detection toolbox for* Kathakali *faces allows virtual shot generation (three simulated shots are shown). The identification of face type allows to further understand the scene and aid video editing.*

will also allow the development of novel visualization tools for better appreciation of the dance form. Style-transfer methods will be useful for pedagogy, where an actor can transfer his performance to an animated *Kathakali* face. Such a framework will be even more important in these difficult COVID times, where physical teaching is difficult.

The proposed pipeline will also aid automated video editing of *Kathakali* dance performance (Fig. 2). An automated editing framework would be extremely beneficial for improving the appreciation and coverage of the *Kathakali* dance form. These detections would allow virtual camera simulations [GRG14; MKSG20]. The identification of face type would allow understanding their roles. For instance, *Kathi* makeup is used for arrogant and evil characters, while the *Pacha* makeup is used to portray kings and divine beings. With access to script, such role identification can aid automated camera selection.

The area of facial understanding has received significant attention over the last couple of decades. The three most commonly used tasks on facial analysis are (a) face detection, (b) landmark detection and (c) semantic segmentation. All three problems have seen tremendous success post the onset of convolutional neural networks. They are used in wide variety of applications like human-computer interaction, head pose estimation [GTGN19], lipsync [CZ16; KSK*17], facial expression editing [WZLC20], virtual makeup [XDZ13] etc. However, these frameworks designed for normal faces fail on *Kathakali* videos because of the presence of elaborate makeup, costume and face masks (Fig. 1 illustrated how the existing state of the art face analysis pipelines fail on *Kathakali* videos).

In this work, we propose a novel toolbox customized for the analysis of *Kathakali* videos. The toolbox covers three tasks, namely: face detection, landmark detection and face segmentation. In the toolbox, we also provide a visualization tool allowing the transfer of *Kathakali* videos onto a cartoon face. We do not claim any architectural novelty; the goal of this work is to demonstrate that a small dataset is sufficient for fine-tuning and customizing existing face analysis methods for the given task. Overall, our paper makes the following contributions:

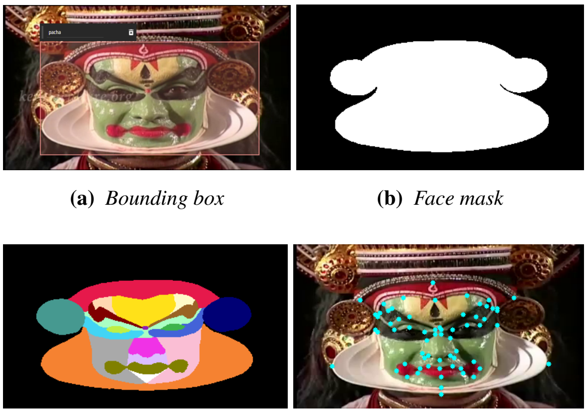
- We present a novel dataset with manually annotated *Kathakali* images. The dataset consists of 1734 images for detection, 148 images for landmark detection and 81 images for face segmentation.
- We show that, albeit small, the dataset suffices to adapt the existing normal face analysis methods onto *Kathakali* faces. We present qualitative and quantitative results to show the efficacy of the adapted methods.
- We adapt GauGAN [PLWZ19] for style-transfer of face segmentation's of a given *Kathakali* sequence onto a *Kathakali* cartoonized template face. We develop a novel data pipeline for curating the paired data for training the GauGAN architecture.

2. Related Work

Kathakali actors narrate the story using *Mudras* (hand gestures) and *Bhavas* (facial-expressions). Previous explorations focused on understanding and classifying *Mudras* displayed by the performer [BI20]. The work of [BI20] classifies the hand gestures into the fixed set of *mudras* on a manually curated dataset, collected in a controlled setting. Our work focuses on the other half of the performance, namely *Bhavas*: the facial expressions.

Analyzing expression on human faces is a well-studied problem. Various datasets and methods have been proposed over the years for tasks like face detection, segmentation and landmark detection. We list a few widely used frameworks and toolboxes:

- **MTCNN** [ZZLQ16] uses unified cascaded CNNs for detecting face bounding boxes and localizing five landmark points on the face.
- **RetinaFace** [DGV*20] gives scale-invariant face localization of human faces via multi-task learning. It does a pixel-wise face localization and gives sparse keypoints along with the detected face. It is extremely robust even if the faces are very-small or partially-occluded.
- **MediaPipe** is an easy-to-use comprehensive toolbox that gives bounding boxes for both the face and the eyes. It also gives sparse key-points, hair-segmentation, eye-gaze detection and face-mesh on human faces accumulating various recent works. [BKV*19; TKV*19; KAGG19; AVG*20].
- **OpenFace 2.2.0** is a another toolbox similar to MediaPipe, based on [BZLM18; ZCBM17; BRM13; WBZ*15; BMR15]. Given a face it performs multiple tasks like face detection, landmark detection, face pose estimation, eye gaze detection and face completion.
- **HRNetV2** [SXLW19; WSC*21; CXW*20] is a library that performs different tasks namely facial landmark detection, face detection, human pose estimation, object detection, semantic and instance segmentation.



(a) *Bounding box* **(b)** *Face mask*

(c) *Region mask* **(d)** *Landmarks*

Figure 3: *Dataset Annotations*

These models are trained on normal human faces and we observe that their performance do not transfer onto *Kathakali* faces, expectedly so, due to the domain shift. We adapt these methods on *Kathakali* domain, by fine-tuning human face models on the proposed dataset. We fine-tune following networks for the respective tasks:

(a) Face Detection - YOLO-V5 [Joc20]
(b) Face Segmentation - DeepLabV3+ [CZP*18]
(c) Landmark Detection - HRNetv2 [SXLW19; WSC*21]

Our work is also related to style transfer methods. We are interested in a specific form of conditional image synthesis, which is converting a semantic segmentation mask to cartoonized *Kathakali* faces. Our work is in tune with other seminal works [IZZE17; CK17] converting segmentation maps to a photorealistic image. We use the GauGAN [PLWZ19] model for our experiments.

3. Dataset

Kathakali is a unique dance-drama art-form that originated in the southern state of *India* and has mostly been unexplored by modern multimedia technology. It has been compared to other popular art forms like a ballet, a miracle play, a dance-drama, an opera and a pantomime. The actor's makeup demarcates the type of character in this dance-drama performance. Our work focuses on the most recurring character, namely *Pacha*. The faces are painted green with large black markings around their eyes and eyebrows, as can be seen in figure 1. To the best of our knowledge, there has not been a prior effort towards creating such a dataset.

We propose a dataset with four different types of annotations. All 1734 images are annotated with bounding box coordinates for face detection (an example ground truth annotation is shown in Fig. 3a). The bounding box is labelled with the type of face detected namely *Pacha* (protagonist) or *Kathi* (antagonist). We collect the images from different image/video search engines like Flickr and Youtube. Our work limits to *Pacha* and *Kathi veshams* (make-up) and others like *Thaadi*, *Minukku*, *Kari* are not covered in the current effort.

We annotate eighty-one images containing *Pacha* in the dataset with face region segmentation. We annotate every image with a binary mask, giving a value zero to non-face pixels (an example ground truth annotation is shown in Fig. 3b). This annotation is similar to the segmentation of human faces in [BPD13]. This annotation can be used to train a model to extract the face region from the image. The eighty-one images used for these annotations were carefully chosen such that all the *bhavas*(expressions) are represented in the dataset.

For the task of semantic segmentation, we annotate the same eighty-one face images with 21 different regions. The regions are chosen such that, during movement of the face, the region within each segment can be approximated by an affine transform. An example ground truth annotation is shown in Fig. 3c.

We further annotate the 148 images with sixty-eight landmarks. The key-points are chosen such that they form the corner points of the regions segmented in the above section. Fig. 3d shows a sample image with landmark annotations. The 148 images consist of the eighty-one images that were annotated with segmentation.

All the images in the dataset for segmentation and landmark detection are frontal or near-frontal faces. Images for face detection include diverse poses. Table 1 summarizes the details of our annotated dataset. We have used the Computer Vision Annotation Tool (CVAT) [Int22] and MakeSense AI [Ska19] to annotate our dataset.

4. Model and Implementation Details

In this section, we explain each of the individual models proposed in our work, namely face detection, face segmentation, landmark detection and semantic segmentation. We fine-tune the models trained for regular human faces on the proposed *Kathakali* dataset. To validate the usefulness of the proposed models in the toolbox, we present a novel application of expression transfer of *Kathakali* videos onto a cartoon template. Fig. 4 shows the architecture diagram of the overall visualization pipeline. It transfers the expression *Bhava* of the *Kathakali* artist onto a reference sketch image. The transfer of expression from a source photorealistic image to the target reference sketch heavily relies on the sub-tasks. We train the model for 4000 epochs with a batchsize of 14 on 1 GPU (RTX 2080Ti).

4.1. Face Detection

Analyzing *Kathakali* performance in the wild requires detection of faces in the video frames. We fine-tune YOLOv5 [Joc20], pre-trained on MS-COCO dataset [LMB*14] (COCO128).

We fine-tune the model for detecting two of the most important *veshams* in the performance: *Pacha* and *Kathi*. These images capture variation in poses, lighting conditions, scales and resolutions. The YOLO family of object detection models use a compound loss calculated based on the object score, class probability score and bounding box regression score. We use binary cross-entropy loss to compute class probability score and object score. To compute loss for bounding box regression, we use CIoU (Complete IoU) loss [ZWL*20].

Tasks	#images	Annotation	*Vesham*(character)	Scope
Face Detection	1734	B-box Coord.	*Pacha*, *Kathi*	in the wild
Face Segmentation	81	Binary mask	*Pacha*	frontal faces
Region Segmentation	81	21 segments	*Pacha*	frontal faces
Landmarks Detection	148	68 key points	*Pacha*	frontal faces

Table 1: *Dataset characteristics*

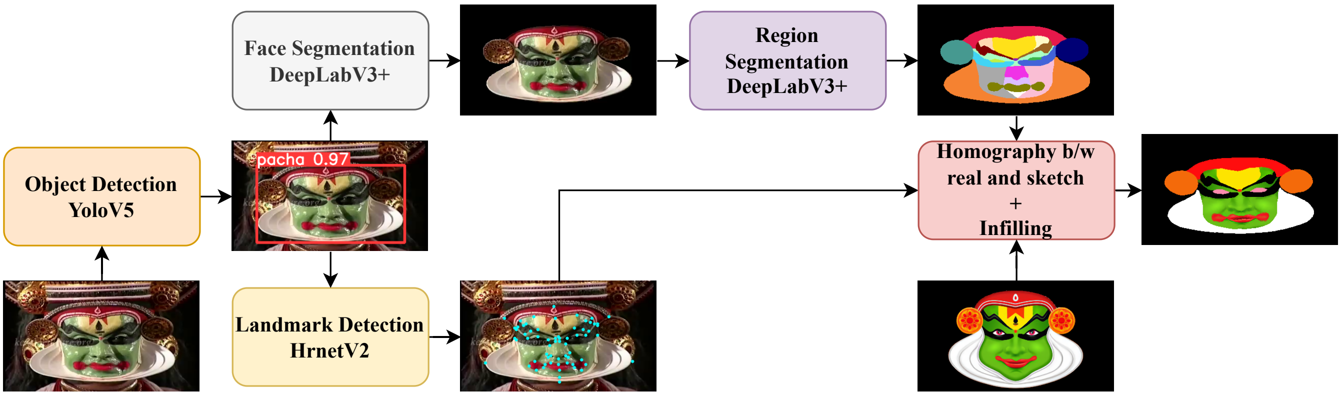

Face Segmentation DeepLabV3+
Region Segmentation DeepLabV3+
pacha 0.97
Object Detection YoloV5
Homography b/w real and sketch + Infilling
Landmark Detection HrnetV2

Figure 4: *The figure shows the architecture of the proposed method. The input is an image from the video, and the output shows the* bhava *(expression) of the actor on a reference sketch image. The pipeline includes five segments. Given an image from the* Kathakali *video, we detect the* Pacha *face with a bounding box. Respective modules give sixty-eight landmark points on the face and twenty-one face segments. The segments and landmarks are used to individually warp each of the segments in the reference image onto the detected segments. We post-process to transfer the* bhava *onto the reference image.*

4.2. Face Segmentation

Post face detection, we extract the face pixels from within the bounding box. We use annotations of segmentation masks of the eighty-one images in the *Kathakali* dataset and fine-tune DeepLabV3+ model [CZP*18] pre-trained on Cityscapes dataset [COR*16]. We train the model with the ground truth binary masks (e.g. Fig. 3b). Our implementation of DeepLabV3+ is based on the mmsegmentation: open-mmlab [Con20]. We augment the dataset with random crops and photo-metric distortions. We normalize the images using the mean and standard deviation of the dataset, i.e. [123.675, 116.28, 103.53] and [58.395, 57.12, 57.375], respectively. We zero pad the images to maintain uniform size post augmentations.

4.3. Semantic Segmentation

After subtracting the background pixels from the face image, we segment the face into twenty-one sub-regions. The regions include nose, mouth, moustache, eyebrows etc (as illustrated in Fig. 3c). Similar to the previous subsection, we fine-tune a DeepLabV3+ model [CZP*18] for this task which is pre-trained on the Cityscapes dataset. The resolution of the input image is 384x216. We again use the mmsegmentation package [Con20] for the task.

4.4. Landmark Detection

We fine-tune hrnetv2 [SXLW19; WSC*21] facial landmark detection model, pre-trained on WFLW dataset [WQY*18], to detect the landmarks on the *Pacha* faces. We use the 148 images with the landmark annotations in the proposed *Kathakali* dataset (e.g. Fig. 3d). The input image has a resolution of 384x216. The model is trained for 1000 epochs, keeping the batch size as 16. Adam optimizer is used with the initial learning rate of 0.0001. The model uses MSE (Mean Squared Error) loss function.

4.5. Style Transfer

We perform conditional image synthesis, i.e., convert a semantic segmentation mask to cartoonized *Kathakali* face. The overall pipeline for the same is illustrated in Fig. 4 and Fig. 5. We aim to use Conditional Generative Adversarial Networks for a segmentation map to cartoon face conversion. We first curate paired data of segmentation maps and corresponding cartoonized faces to facilitate the same.

Paired dataset curation: We first set correspondences between landmarks on a realistic and cartoonized *Kathakali* face. Given a *Kathakali* frontal face, we detect landmarks and perform semantic image segmentation on it. We then use the predicted landmark points to compute homography transform between the segmentation map and the template cartoonized face (Fig. 6b). Using this homography matrix, we individually warp each of the face segments of the cartoonized template image to fit the corresponding

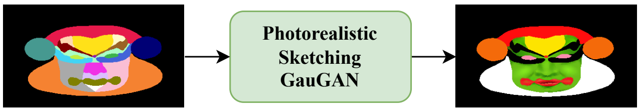
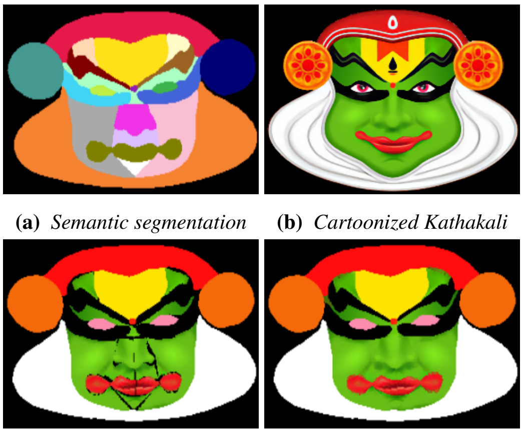


Figure 5: *The proposed pipeline generates a segmentation mask and cartoonized* Kathakali *template image for each frame of the video. We train a GauGAN over this segmentation and template image pair to get a consistent visualization of the transferred* bhava.



(a) *Semantic segmentation* **(b)** *Cartoonized Kathakali*

(c) *After warping* **(d)** *After infilling*

Figure 6: *Face Warp output*

segmentation map. During the warping process, some of the regions in Figure 3 are further broken down into smaller segments to preserve the shape. Fig. 6 shows the output of this step.

The output post-warping creates uneven gaps because we assume that each segment moves independently. We post-process this image to produce a smooth template figure using the Navier-Stokes inpainting method [BBS01]. Fig. 6d illustrates the inpainted output. The images Fig. 6a and Fig. 6d form a pair.

The paired dataset curation pipeline (Figure 4) can be directly used for style transfer. However, we find that the noise in the landmark detection process leads to significant temporal flicker. For this reason, we train a GauGAN model [PLWZ19] for conditional image synthesis. GauGAN takes a segmentation mask as input and generates a photorealistic image as output (as illustrated in Figure 5). The model is trained for 50 epochs on 2 GPUs (RTX 2080Ti), keeping the batch size as 8. Adam optimizer is used with the initial learning rate of Generator as 0.0001 and that of Discriminator as 0.0004.

5. Results

In this section, we present the results for each of the five modules introduced in Section 4. While training each of the modules, we keep aside 15% of the images for testing. We report results for each task on test images.

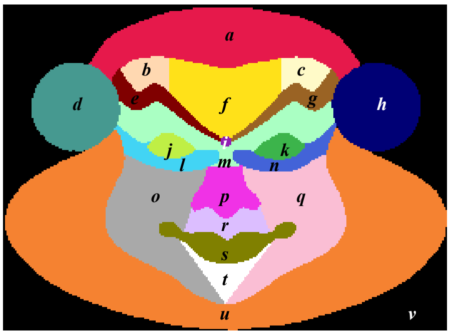


Figure 7: *The figure shows the labels of each segment used in the Semantic Segmentation task*

Face Detection: The output of the face detection module is a bounding box corresponding to each *Kathakali* face in the input image. A label and a confidence score are also predicted corresponding to each of the bounding boxes. The labels are either *Pacha* or *Kathi*. The model achieves a mean Average Precision (mAP@0.5) score of 0.997 for the label *Pacha* and 0.996 for the label *Kathi* giving an overall mAP@0.5 score of 0.997. It is interesting to note that this score is comparable to the mAP@0.5 score of the model on human face datasets. Fig. 8a and fig. 2 shows the output of face detection on one of the frames.

Face Segmentation: The model achieves a mean Intersection over Union (mIoU) score of 0.9847 and mean Accuracy of 99.27% over the validation data. Table 3 shows the segmentation scores for both background and face regions. Fig. 8b shows the output of face segmentation.

Semantic Segmentation: The model achieves a mIoU score of 0.81 and mAcc score of 89.02% across all segments in the validation data. The Table 2 gives the mIoU and mAcc scores for each of the twenty one regions along on the validation data. Fig. 8c shows the output of face segmentation.

Landmark detection:The model achieves normalized mean error score of 0.0175 over the validation data. Fig. 8d shows the output of landmark detection on nine-emotions (in *Kathakali* the nine emotions are called the *Navarasas*).

Expression transfer: Figure 6c shows the gaps and overlaps created while warping the individual segments from the cartoon template image 6b onto the photo-realistic image. We fill these gaps using a combination of direct colours and Navier-Stokes infilling. Fig. 6 shows the image post the infilling operation. We call the output image of this operation as $template_{emot}$.

The pipeline shown in Figure 4 gives template sketch replicating the *Bhava* (expression) of the *Kathakali* artist in the input frame. The consistency (both spatial and temporal) of the output are highly dependent on the accuracy and precision of the individual components. For instance, an aberration in the position of the predicted landmark can create undesired artefacts in the output. We use our pipeline to generate data and train a GauGAN using segmentation mask 6a and $template_{emot}$ 6d pair. At inference, the frames from *Kathakali* video and the corresponding segmentation mask is used as input, and GauGAN predicts $template_{emot}$. The input and output

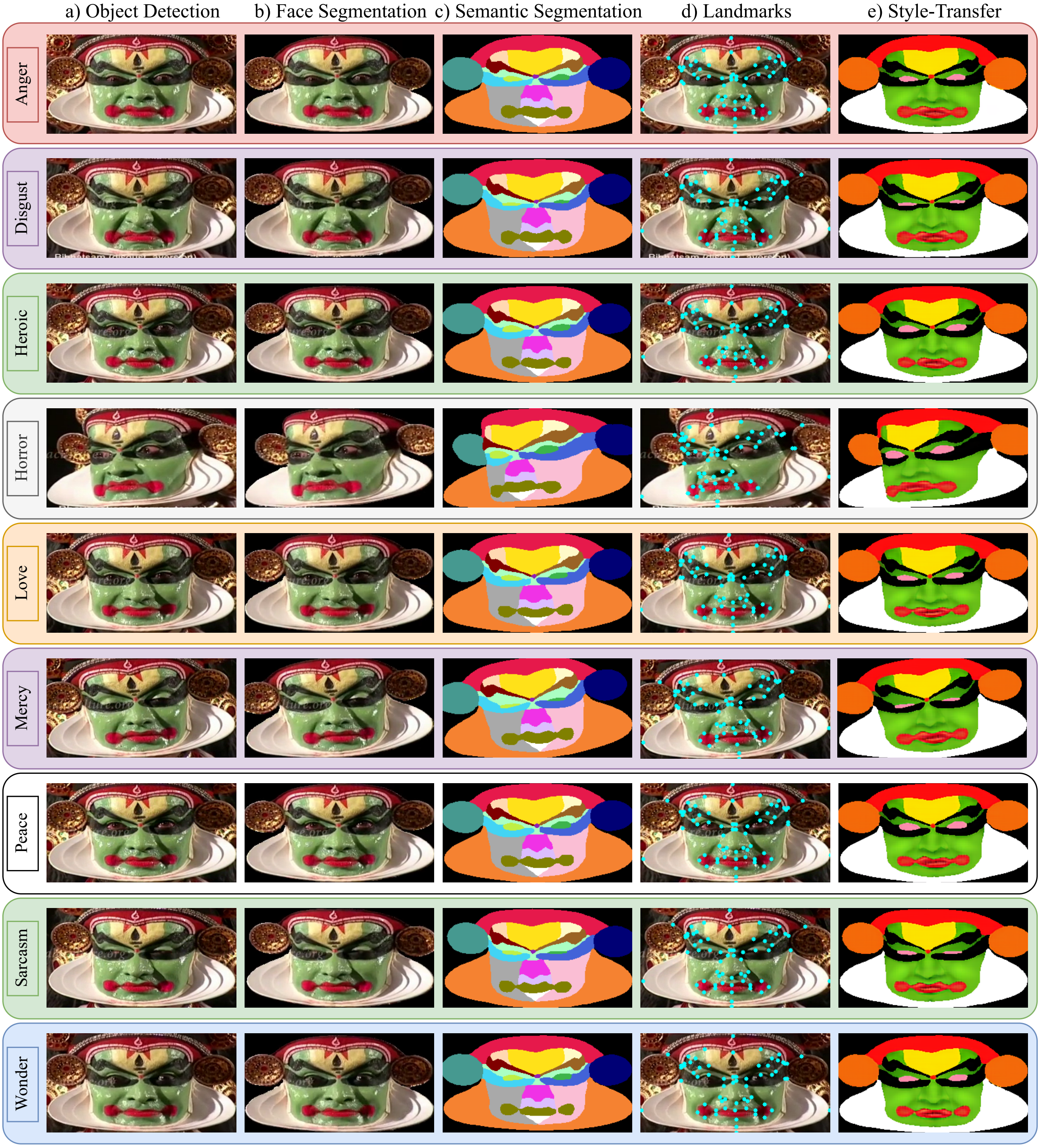


Figure 8: *Final Result of Cartoonized(style-transfer)(right) and intermediate results of the pipeline(Face Segmentation, Semantic Segmentation, Landmark Detection)*

Label	Region	IoU	Acc
a	red_forehead	0.9206	96.03
b	green_forehead_left	0.8462	91.66
c	green_forehead_right	0.7819	89.06
d	left_ear	0.9297	97.83
e	left_eyebrow	0.8299	89.48
f	yellow_forehead	0.9187	97.29
g	right_eyebrow	0.7193	81.49
h	right_ear	0.9238	97.93
i	eyebrow_center	0.4938	63.43
j	right_eyeball	0.7243	84.49
k	left_eyeball	0.6617	77.27
l	left_eye_makeup	0.8191	90.31
m	between_eyes_eyebrows	0.6998	83.63
n	right_eye_makeup	0.8056	89.42
o	left_cheek	0.8832	94.04
p	nose	0.8591	92.03
q	right_cheek	0.8622	92.03
r	moustache	0.7630	85.89
s	mouth	0.8179	90.59
t	chin	0.6640	78.12
u	beard	0.9202	96.97
v	background	0.9913	99.37

Table 2: *Metric value for each individual region of the Semantic Segmentation. The regions are labelled in fig. 7*

Region	IoU	Acc
background	0.9928	99.60
outer_face	0.9766	98.93

Table 3: *Metric value for each individual region of the Face Segmentation*

samples are shown in Figure 5. We observe this output to be more consistent.

We compare the proposed GauGAN model with CycleGAN [ZPIE17] and Articulated Animation [SWR*21] in Figure 9. Articulated Animation [SWR*21] and CycleGAN, both are unsupervised approaches and only use a single cartoon sketch for training. We can observe that both these method fail to generate consistent translations. We can also observe errors with change in pose. The experiment clearly suggests the efficacy of the proposed method and highlights the importance of the novel paired data generation pipeline.

6. Conclusions

Kathakali is a classical Indian dance form which has been relatively unexplored by multimedia technology. AI models trained on human faces fail to perform on *Kathakali* faces as the performer wears an elaborate and colourful makeup, costume and face mask. We present a dataset specific to *Kathakali*, with four manual annotations, namely face detection, background subtraction, landmark detection and face semantic segmentation. We show that the existing models trained on normal human faces adapt to the *Kathakali* faces with the proposed data. We make available modules for the aforementioned tasks on *Kathakali* faces and show the use of the modules by visualizing the expressions of a *Kathakali* artist on a cartoon template *Kathakali* image. We train a generative model using this data and show a temporally consistent transfer of expression of a performer to the cartoon *Kathakali* template image. We believe the proposed toolkit will help better the learning and appreciation of the classical art form.

7. Acknowledgement

This work was supported under IHDS cluster project by Department of Science & Technology, Government of India.

We would like to thank Rohan Reddy for helping modify the template image, Ritwik Agarwal and Harshdeep Singh for assisting in landmarks annotation, S. Dhawal for improving the aesthetics of the figures.

References

[AVG*20] Ablavatski, Artsiom, Vakunov, Andrey, Grishchenko, Ivan, et al. "Real-time Pupil Tracking from Monocular Video for Digital Puppetry". *arXiv:2006.11341* (2020) 2.

[BBS01] Bertalmio, Marcelo, Bertozzi, Andrea L, and Sapiro, Guillermo. "Navier-stokes, fluid dynamics, and image and video inpainting". *CVPR*. 2001 5.

[BI20] Bhavanam, Lakshmi Tulasi and Iyer, Ganesh Neelakanta. "On the classification of Kathakali Hand Gestures using support vector machines and convolutional neural networks". *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*. 2020 2.

[BKV*19] Bazarevsky, Valentin, Kartynnik, Yury, Vakunov, Andrey, et al. "Blazeface: Sub-millisecond neural face detection on mobile gpus". *arXiv:1907.05047* (2019) 2.

[BMR15] Baltrušaitis, Tadas, Mahmoud, Marwa, and Robinson, Peter. "Cross-dataset learning and person-specific normalisation for automatic action unit detection". *FG*. 2015 2.

[BPD13] Burgos-Artizzu, Xavier P., Perona, Pietro, and Dollár, Piotr. "Robust Face Landmark Estimation under Occlusion". *ICCV*. 2013 3.

[BRM13] Baltrusaitis, Tadas, Robinson, Peter, and Morency, Louis-Philippe. "Constrained local neural fields for robust facial landmark detection in the wild". *ICCV workshops*. 2013 2.

[BZLM18] Baltrusaitis, Tadas, Zadeh, Amir, Lim, Yao Chong, and Morency, Louis-Philippe. "Openface 2.0: Facial behavior analysis toolkit". *FG*. 2018 2.

[CK17] Chen, Qifeng and Koltun, Vladlen. "Photographic image synthesis with cascaded refinement networks". *ICCV*. 2017, 1511–1520 3.

[Con20] Contributors, MMSegmentation. *MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark*. https://github.com/open-mmlab/mmsegmentation. 2020 4.

[COR*16] Cordts, Marius, Omran, Mohamed, Ramos, Sebastian, et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". *CVPR*. 2016 4.

[CXW*20] Cheng, Bowen, Xiao, Bin, Wang, Jingdong, et al. "HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation". *CVPR*. Mar. 2020 2.

[CZ16] Chung, Joon Son and Zisserman, Andrew. "Out of time: automated lip sync in the wild". *ACCV*. 2016 2.

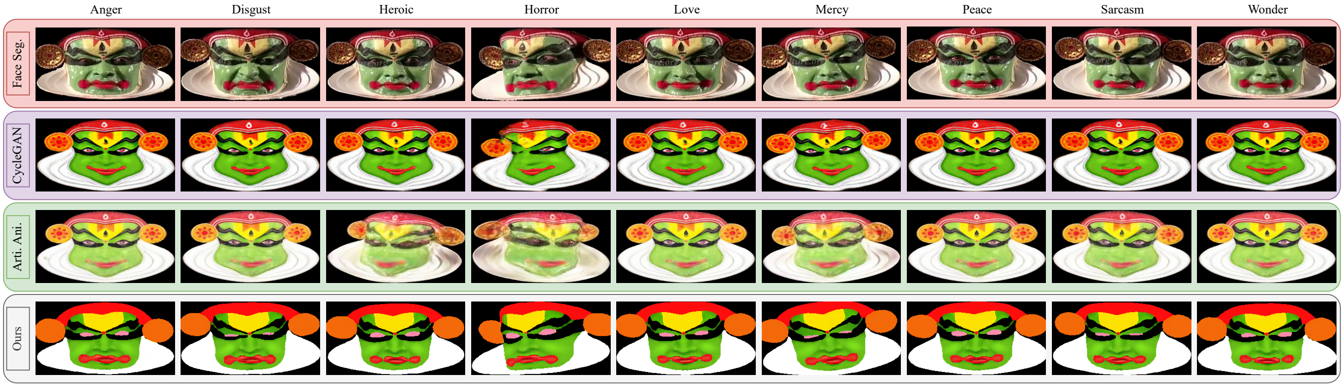


Figure 9: *Comparing the proposed GauGAN model (last row) with CycleGAN [ZPIE17] (second row) and Articulated Animation [SWR*21] (third row)*

[CZP*18] Chen, Liang-Chieh, Zhu, Yukun, Papandreou, George, et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation". *ECCV*. 2018, 801–818 3, 4.

[DGV*20] Deng, Jiankang, Guo, Jia, Ververas, Evangelos, et al. "Retinaface: Single-shot multi-level face localisation in the wild". *CVPR*. 2020 2.

[GRG14] Gandhi, Vineet, Ronfard, Remi, and Gleicher, Michael. "Multi-Clip Video Editing from a Single Viewpoint". *European Conference on Visual Media Production (CVMP)*. 2014 2.

[GTGN19] Gupta, Aryaman, Thakkar, Kalpit, Gandhi, Vineet, and Narayanan, PJ. "Nose, eyes and ears: Head pose estimation by locating facial keypoints". *ICASSP*. 2019 2.

[Int22] Intel. *Powerful and efficient Computer Vision Annotation Tool (CVAT)*. https://github.com/openvinotoolkit/cvat. 2022 3.

[IZZE17] Isola, Phillip, Zhu, Jun-Yan, Zhou, Tinghui, and Efros, Alexei A. "Image-to-image translation with conditional adversarial networks". *CVPR*. 2017 3.

[Joc20] Jocher, Glenn. *ultralytics/yolov5*. https://github.com/ultralytics/yolov5. Version v3.1. 2020 3.

[KAGG19] Kartynnik, Yury, Ablavatski, Artsiom, Grishchenko, Ivan, and Grundmann, Matthias. "Real-time facial surface geometry from monocular video on mobile GPUs". *arXiv:1907.06724* (2019) 2.

[KSK*17] Kumar, Rithesh, Sotelo, Jose, Kumar, Kundan, et al. "Obamanet: Photo-realistic lip-sync from text". *arXiv* (2017) 2.

[LMB*14] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, et al. "Microsoft coco: Common objects in context". *ECCV*. 2014 3.

[MKSG20] Moorthy, KL Bhanu, Kumar, Moneish, Subramanian, Ramanathan, and Gandhi, Vineet. "Gazed–gaze-guided cinematic editing of wide-angle monocular video recordings". *CHI*. 2020 2.

[PLWZ19] Park, Taesung, Liu, Ming-Yu, Wang, Ting-Chun, and Zhu, Jun-Yan. "Semantic Image Synthesis with Spatially-Adaptive Normalization". *CVPR*. 2019 2, 3, 5.

[Ska19] Skalski, Piotr. *Make Sense*. https://github.com/SkalskiP/make-sense/. 2019 3.

[SWR*21] Siarohin, Aliaksandr, Woodford, Oliver J., Ren, Jian, et al. "Motion Representations for Articulated Animation". *CVPR*. June 2021, 13653–13662 7, 8.

[SXLW19] Sun, Ke, Xiao, Bin, Liu, Dong, and Wang, Jingdong. "Deep High-Resolution Representation Learning for Human Pose Estimation". *CVPR*. 2019 2–4.

[TKV*19] Tkachenka, Andrei, Karpiak, Gregory, Vakunov, Andrey, et al. "Real-time hair segmentation and recoloring on mobile gpus". *arXiv:1907.06740* (2019) 2.

[WBZ*15] Wood, Erroll, Baltrusaitis, Tadas, Zhang, Xucong, et al. "Rendering of eyes for eye-shape registration and gaze estimation". *ICCV*. 2015 2.

[WQY*18] Wu, Wayne, Qian, Chen, Yang, Shuo, et al. "Look at Boundary: A Boundary-Aware Face Alignment Algorithm". *CVPR*. 2018 4.

[WSC*21] Wang, Jingdong, Sun, Ke, Cheng, Tianheng, et al. "Deep High-Resolution Representation Learning for Visual Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.10 (2021), 3349–3364 2–4.

[WZLC20] Wu, Rongliang, Zhang, Gongjie, Lu, Shijian, and Chen, Tao. "Cascade ef-gan: Progressive facial expression editing with local focuses". *CVPR*. 2020 2.

[XDZ13] Xu, Lin, Du, Yangzhou, and Zhang, Yimin. "An automatic framework for example-based virtual makeup". *ICIP*. 2013 2.

[ZCBM17] Zadeh, Amir, Chong Lim, Yao, Baltrusaitis, Tadas, and Morency, Louis-Philippe. "Convolutional experts constrained local model for 3d facial landmark detection". *ICCV Workshops*. 2017 2.

[ZPIE17] Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, and Efros, Alexei A. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks". *ICCV*. 2017 7, 8.

[ZWL*20] Zheng, Zhaohui, Wang, Ping, Liu, Wei, et al. "Distance-IoU loss: Faster and better learning for bounding box regression". *AAAI*. 2020 3.

[ZZLQ16] Zhang, Kaipeng, Zhang, Zhanpeng, Li, Zhifeng, and Qiao, Yu. "Joint face detection and alignment using multitask cascaded convolutional networks". *IEEE signal processing letters* 23.10 (2016), 1499–1503 2.