Why Don't You Speak?: A Smartphone Application to Engage Museum Visitors Through Deepfakes Creation

Matteo Zaramella
zaramella.2025806@studenti.uniroma1.it
Sapienza University of Roma
Rome, Italy

Irene Amerini
amerini@diag.uniroma1.it
Sapienza University of Roma
Rome, Italy

Paolo Russo
prusso@diag.uniroma1.it
Sapienza University of Roma
Rome, Italy

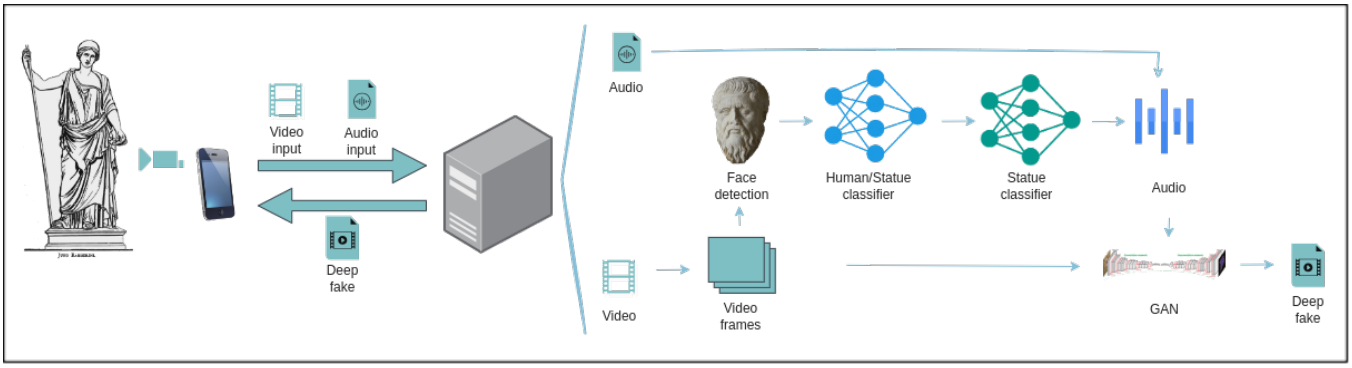

Figure 1: The pipeline of the *Why don't you speak?* application. On the left the Android application and on the right the Server application with evidenced the different parts of the pipeline.

ABSTRACT

In this paper, we offer a gamification-based application for the cultural heritage sector that aims to enhance the learning and fruition of museum artworks. The application encourages users to experience history and culture in the first person, based on the idea that the artworks in a museum can tell their own story, thus improving the engagement of the museums and providing information on the artwork itself.

Specifically, we propose an application that allows museum visitors to create a deepfake video of a sculpture directly through their smartphone. More in detail, starting from a few live frames of a statue, the application generates in a short time a deepfake video where the statue talks by moving its lips synchronized with a text or audio file. The application exploits an underlying generative adversarial network technology and has been specialized on a custom statues dataset collected for the purpose. Experiments show that the generated videos exhibit great realism in the vast majority of cases, demonstrating the importance of a reliable statue face detection algorithm. The final aim of our application is to make the museum experience different, with a more immersive interaction and an engaging user experience, which could potentially attract more people to deepen classical history and culture.

SUMAC '23, November 2, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0279-2/23/11…$15.00
https://doi.org/10.1145/3607542.3617359

CCS CONCEPTS

• **Computing methodologies** → *Computer vision.*

KEYWORDS

museum user experience, deepfake, face detection, generative adversarial network

ACM Reference Format:
Matteo Zaramella, Irene Amerini, and Paolo Russo. 2023. Why Don't You Speak?: A Smartphone Application to Engage Museum Visitors Through Deepfakes Creation. In *Proceedings of the 5th Workshop on the analySis, Understanding and proMotion of heritAge Contents (SUMAC '23), November 2, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3607542.3617359

1 INTRODUCTION

The technique of incorporating tactics and game dynamics into nongame environments is known as gamification [23]. It has previously been demonstrated to be helpful for improving abilities in several fields, including marketing, business training, and entertainment. A gamification strategy may undoubtedly enhance cultural heritage by providing opportunities to engage visitors with the content of museums through the design of more enjoyable and challenging digital learning scenarios [14].

Why don't you speak? is a new application to engage museum visitors allowing them to interact with historical figures through audio and video messages. The proposed solution would enhance the cultural and educational value of a sculpture, as well as provide entertainment and engagement for the audience. An active interaction between visitors and the statue is promoted, as opposed to traditional description texts which could lead the user to distraction

and boredom. Based on the notion that the artworks at a museum may tell their own story, boosting the engagement of the museums, and offering information about the artworks themselves, the application encourages users to experience history and culture in the first person. However, creating talking statue videos requires a considerable amount of time, effort, and expertise, as it involves scripting, recording, editing, and animating the statue's face and voice. In order to greatly ease the process of creating talking statue videos, we developed an application that can automatically animate any statue by moving its lips synchronized with an audio or text file. Our solution consists of an Android application (frontend), employed by the users through the smartphone, and a server application (backend) which runs processing algorithms and a deep generative model for deepfake video creation.

We designed our Android application to work in a very simple and user-friendly way: the user takes a short live video of a statue which is sent to the server that, after the application of processing and generation steps, sends the generated deepfake video back to the smartphone, which shows the video to the user. In order to build a set of data for the experiments, we created a dataset of statues by acquiring videos from the Gipsoteca Museum of Sapienza University of Rome [1] (Rome, Italy). In the Gipsoteca ("gypsotheque") is possible to admire plaster casts of Greek sculptures from the Archaic period to Hellenism. In our project, we consider different sculptures (i.e. Poseidon, Hercules, Zeus, Minerva Tritonia, etc.) among the 1200 plaster casts. We demonstrate the power of our approach on the collected dataset, which will be expanded in future works.

2 RELATED WORKS

As far as we know, there is no algorithm or program that can automatically produce talking statue videos. Talking statues videos, however, are similar to lip sync videos, which are usually created using software like DeepFaceLab [18] [4] and its live variant DeepFaceLive [5]. Such applications are based on the face swap mechanism [15], a specific task belonging to the broader research field called AI-generated media like deepfake [16], and are very diffused due to their output quality and simplicity of use. Moreover, other models generating deepfake using the method of swapping the faces have been recently developed, like *faceswap* [7], which can be considered an updated version of *DeepFaceLab*.

Another possibility to integrate video and audio is to use the method proposed by Prajwal et al.[19] called *Wav2Lip*. It is a popular, state-of-the-art lip sync model which exploits a Generative Adversarial Network for creating a video of a human face in which the lips are moving synchronized with an audio file. Due to the high quality and realism of the generated videos, we started from this algorithm to create the model for producing talking statues videos. Another way to do it, is presented on "Perceptual Conversational Head Generation with Regularized Driver and Enhanced Renderer" [12]. This projct aims to generate a face to face conversation based on an audio and the reference images. It modifies the faces to make people talk and use also the foreground-background fusion technique to accentuate the movement. It obtains very good results and got first place in the listening head generation track on ACM Multimedia ViCo 2022 Conversational Head Generation Challenge. Regarding the field of video generation in general, a similar function can be performed with the Singer et al. work [24], developed by Meta AI group. The authors propose a technique able to create a video starting from an image, text or another video fed as input. The model implements a variation of the Text-to-image paradigm so that any possible input is firstly converted into a proper embedding vector, which is then exploited for the video creation thanks to the captured input semantics. One more video generation project is proposed by Zhuo Chen et al. [10], which propose a pipeline to create a video starting from two images. It is based on styleGAN2 and is implemented in two step, the CluDistiller framework which exctacts poses and expressions and convert them in the latent space, and the AugDistiller that merge the extracted expressions to the input images. Another interesting work, still based on styleGAN but the version 3, is the generative model called DraGAN [17]. It is able to take as input an image, modify it and create a new, different image as output. It can be considered one of the state-of-the-art models for deepfake on images. A different approach is adopted by Pumarola et al. [20], which aims to animate the faces adding emotions. In fact, starting from an image, it propose a Generative Adversarial Network called GANImation that add feelings to the image modifying the muscles of the face. One more work related to generative model is the MirrorGan [21] that aims to generate images from text and then convert back the image to text (Text to Image and Image to text), this is considered state of the art on its field. It starts embedding the text and generating the image using a Generative Adversarial Network and then, from the image, trys to reconstruct the text using a serie of LSTMs. Finally, talking about smartphone applications our work can be related to the *Talking Statues* Android app [11] which, is simply an app for running predefined audio files on statues that support this application with a QR code.

3 SYSTEM DESIGN

Our solution, whose pipeline is described in Figure 1, can be split into two different parts: an Android application constituting the frontend, used by the visitors, and a server application (the backend) which runs processing algorithms and a deep generative model for video creation. The application records the live video, and then sends it to the server application. The server application analyzes the received video and generates the deep fake. Once the deep fake is ready, it is sent back to the Android application, and it is shown to the user.

3.1 Android Application

The frontend application is meant to be used by the visitors of the museum through the use of an Android smartphone (for this project the visitors of Gipsoteca Museum at Sapienza University of Rome). The application is developed in Java language using Android Studio with the use of Camera2 [3] API to access the camera and HttpURLConnection API [9] to interact with the server. The application starts with a splash screen activity and then appears a menu with three functionalities (see Figure 2 on the left). In the following, each of them is explained in detail, while Figure 2 shows the app instructions to fulfill each functionality:

[1] https://web.uniroma1.it/polomuseale/en/node/5653

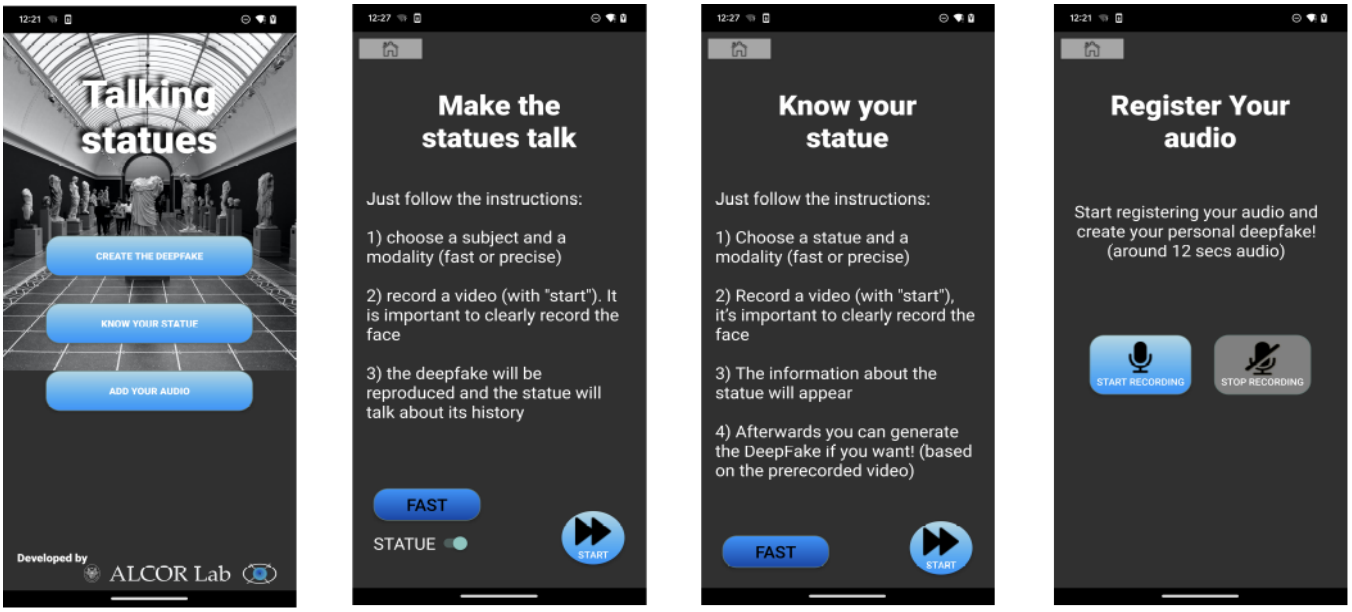

Figure 2: The screenshots of the *Why don't you speak* App for smartphone (from left to right Home page, Create the deepfake, Know your statue and Add your audio).

(1) *Create the deepfake*: the user records a small video of 10-15 seconds, recording the statue's face; then, a deepfake video is created in the backend and shown in the smartphone app. More in detail, the Android application, first of all, records the video with MediaRecorder library [2], then, sends the video through a POST request to the server and waits for an answer. If the response is equal to "Error", it means that the video doesn't contain any recognized face, so the application shows the user an error message and asks him to take another video; otherwise, if the deepfake is correctly generated, it starts playing the generated deepfake to the user. The video will explain details regarding the sculpture framed by the user by making the statues talk (moving lips) and by reproducing the audio file associated with the statue. Once the video ends, the visitor can choose to rewatch it, save it in his or her gallery (allowing the user to bring the museum experience home), or go back to the main menu to generate a new deepfake.

(2) *Know your statue*: the user records a short video of the statue's face. The application will send such video to the server where a machine learning classifier will recognize if the video frames are related to a statue's face and will recognize the statue in question. Once the user has finished recording the video, the Android application waits until the server sends back a label containing the name of the statue (if the statue is not recognized as one present in the dataset, the answer is "other"). Once received the name of the statue, the application shows all the related information taken from a dictionary. More in depth, the app accesses two dictionaries, one to retrieve the sculpture description and a second one to get the stored image. In this way, at the end of this process, the user can have access to a textual description of the statue Figure 3. Successively it will be possible to generate the deepfake directly from the just recorded video.

(3) *Add your audio*: the user can record his own audio directly with the smartphone to be associated with the deepfake video, just pressing the button "Start recording". The function of registering the audio is implemented using the MediaRecorder library [2], and once the audio is recorded (and the user decided that the audio is good), it is sent to the server and saved for the deep fake generation. Once the audio is received, the user records a video of the subject/ statue for the creation of the deepfake. Such video is sent to the server application. As soon as the deepfake is generated by the server the deepfake will be shown and played by the app together with the audio.

3.2 Server Application

The server application, which is the backend of our application, has been developed in Python and implemented using the Flask micro web framework [8], the OpenCV library, and the Pytorch framework. The preprocessing module of the server application is composed of a face detector and two classifiers: one devoted to discriminating between human and statue faces and the second one related to the classification of the statue (inferencing the category). After the statue classification is performed, the audio associated with that statue is retrieved and the application starts to generate the deepfake. When the deepfake video is ready it is sent back to the client application and the result is shown to the user.

3.2.1 Face detection and Statue classification. Detecting faces and identifying statues is a crucial step for the application, as it allows

Figure 3: The screenshot of the *Know your statue* functionality applied on the Hercules statue.

to recognize any statue in the video and proceed with the deepfake video creation and the audio matching.
In order to speed up the algorithm, we decided to run this phase by default on the first video frame only, with the idea that the statue is present in the whole video if recognized in the first frame. However, it is possible to switch, with a simple button on the Andorid application, to a more conservative approach, which is to run the same preprocessing pipeline on the whole frame sequence, rejecting the video if the statue is not recognized in one of the frames.

The first step of the backend pipeline is to perform face detection. To do it, we opted for the pre-trained YOLOv8 model [25], which is the latest version of the popular YOLO architecture, which showed very high accuracy on that task. Specifically, this model uses a convolutional neural network to extract features from the input image, then it uses a grid of cells to predict the bounding box for each face in the image. At each step in the inference process, the network produces a set of probability maps that indicate the likelihood that a face is present in a particular location within the image. The location of the highest probability map is used to determine the bounding box for that face. Finally, it returns a list of boxes containing the position and dimension of the bounding box relative to the face found. The list can eventually be empty if there are no faces in the image, or it can have one or more elements if one or more faces are present in the image.

After the first check, if a face is detected, the frame is fed to two classifiers. The first one is a binary Human/Statue classifier (H/S classifier) with the objective to understand if the face in the video belongs to a human or a statue. If the face belongs to a statue, the video is fed to the second classifier which performs a statue classification among the 11 possible statues present in our dataset (plus a 12th category which represents the "other" statue). For the H/S classification an EfficientNet [6] architecture is used. Such CNN is pretrained on ImageNet [1] and finetuned on the specific task. Among all the possible choices, the EfficientNet model represents a good compromise between time execution and accuracy in the detection, with the possibility to switch for heavier but more accurate deep networks at the cost of increased elaboration times.

The Statue Classifier aims to classify the statue present in the frame, in such a way the system is able to select the corresponding audio for the deepfake generation. The model is again an EfficientNet [6] pretrained on ImageNet [1]. The model returns a probability that the given frame is associated with a certain statue. If the probability is below 50%, the image is associated with the class "other" so any statue is detected. Both the H/S and the Statue classifiers have been finetuned on the dataset we collected from the Gipsoteca museum (see Section 4).

3.2.2 Deepfake generator. An essential functionality implemented on the server side is the creation of the deepfake. More in detail, if a statue has been detected and classified by the previous step, the full video is fed to the deepfake generator, together with the audio corresponding to that statue. Among all the possible choices, as a deepfake generator, we opted for the *Wav2Lip* [19] architecture since it demonstrated to be one of the state-of-the-art techniques for lip sync tasks. We modified this architecture by using YOLOv8 [25] as a face detector, which is more accurate and precise than the Multi-Task Cascaded Convolutional Neural Network face detector [26] of the original paper. YOLOv8 face detector algorithm, differently from the previous YOLO versions, is an anchor-free model, which means it predicts directly the center of the detected face instead of the offset from a known anchor box. We applied it on each image of the frame sequences; then, the face bounding box coordinates are used to crop each image and store the cropped face into a tensor. This tensor is duplicated into another tensor while having the bottom half part of the images blackened. Finally, the two tensors together with the audio file (converted into a mel spectrogram [22]) are fed to the Wav2lip Generative Adversarial Network model, which produces the moving lips effect by filling the bottom, blackened part of the image sequence. After this step, the generated faces are copied into the corresponding frames of the original, uncropped video, in order to obtain a realistic deepfake. Finally, the video is merged with the audio file by using the FFMPEG library and sent back to the Android frontend app for visualization.

4 DATASET AND SETUP

We built our application on a statues video dataset which we collected from the Gipsoteca Museum of Sapienza. The statues are placed on pedestals and have a height of between 1.5 and 2 meters. The dataset has been created by taking, for each of the eleven selected statues, an average of 100 photos from each statue with different viewpoints. Moreover, we added three 15-second videos for each statue which served both as additional images for the training set and as test videos for testing the whole pipeline. Once obtained the data, we split them to create a training, validation, and test set. The images have been used for training and testing several processing algorithms (face statue detection, Human/Statues classification,

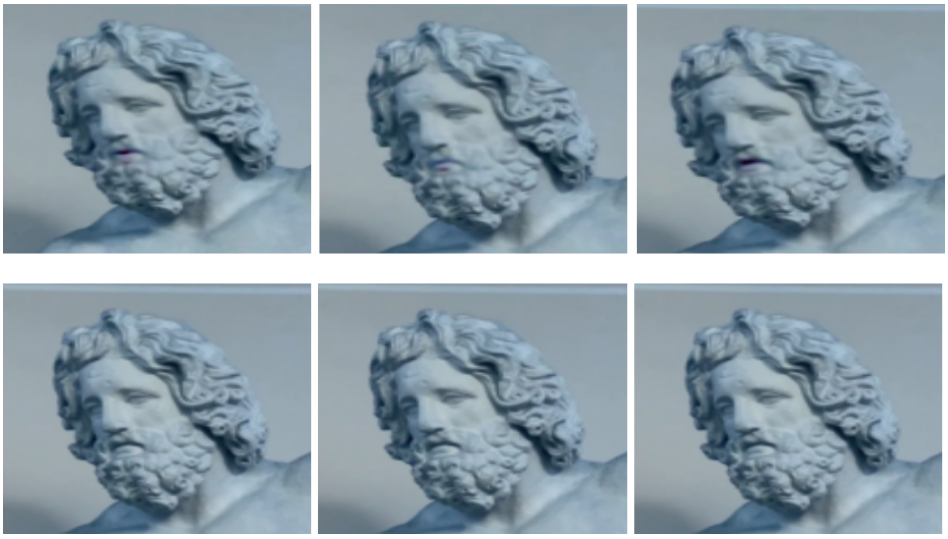

Figure 4: Set of frames related to the video of the statue of Poseidon: three frames from the deep fake videos (first row) and frames from the original video (second row)

and Statues classification) which we have exploited in our application. As previously mentioned, 100 images per statue were not enough to get the best performances possible from the employed classifiers; for this reason, we randomly added some frames from the video to increase the final images dataset. The final training set counts 450 images for each statue, 50 in the validation set and 100 in the test set. Moreover, in order to add further variability, we performed data augmentation (applying random changes in brightness, contrast, saturation, and hue values) generating two additional images for each training image; this helped build a more robust classifier. The final dataset size is reported in Table 1. The full training set has been exploited for training the Statue classifier which recognizes the statue among 12 possible statue labels. Instead, in order to train the Human/Statue binary classifier, we randomly took 500 images from the training set, pairing them with 500 random images of humans taken from Kaggle [13].

Set	photos	video	modified images	Tot img
Training	50	1	100	450
Validation	10	1	0	50
Test	20	1	0	100

Table 1: Training, Validation and Test set employed for the Statues dataset

During the training of classifiers, we utilized the Cross-Entropy Loss as a loss function, Adam as optimizer, with a learning rate of 0.001, and trained for five epochs.
In order to evaluate the performances of the classifiers we used the following metrics:

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Number of total prediction}}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

To record the live video in the Android application, we used, as already mentioned, the MediaRecorder library [2], set the frame rate of acquisition to 30 frames per second to obtain a good quality video, and the output format to .mp4. The chosen video and audio encoders are H264 and AAC respectively. The video encoding bit rate was set to 1000K.

5 EXPERIMENTAL RESULTS

In the following, the quantitative and qualitative results obtained in the different phases of the pipeline are reported.

5.1 Classifications results

The first evaluated experiment regards the accuracy of the face detection algorithm as if no face is detected the whole pipeline will interrupt; for this reason, high accuracy is required. The performances of this module demonstrated to be very good even if the employed network is pre-trained on human faces, with an accuracy value equal to 99% on our statues test set. As a second analysis, we show the results obtained while evaluating the Statue/Human classifier. As already described in Section 4 we built a dataset composed of 1000 random samples for the Training set (500 statues and 500 humans), 110 images are used for the validation set, and 110 for the test set (respectively 55 statues and 55 humans each). In Table 2 is possible to appreciate that the classifier reaches a very high accuracy in distinguishing between human and statue faces. More in detail, less than 2% of the samples have been wrongly classified; the precision score in particular is equal to 1.0, implying that the

number of false positives (statues incorrectly classified as humans) is zero. The same behavior is appreciable from the confusion matrix in Figure 5 which shows the percentages of accuracy, false positives, and false negatives.

Metric	Score
Accuracy	0.981818
F1 Score	0.981481
Precision	1.0
Recall	0.963636

Table 2: Evaluation metrics on the Human/Statue classifier

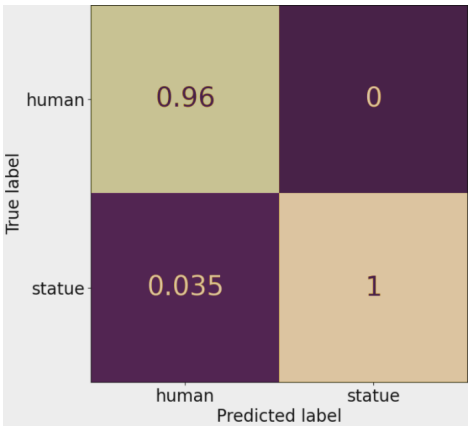

Figure 5: Confusion matrix for the Human/Statue classifier on the test set.

Regarding the Statue classifier, the results are given in Table 3. The model achieves an accuracy of more than 92%, with very high Precision and Recall. From the confusion matrix Figure 6 is possible to notice that for some statues, in particular for the ones with peculiar details in the face (as Hera Barberini reported in Figure 7), the classifier reaches 100% classification accuracy. On the other hand, for some other statues, the classifier reaches a classification accuracy of around 80%, like for Poseidon and Dionysus. We suppose that a contributing factor to this behavior could be the different environmental conditions like illumination and orientation of the statue and characteristics of the face.

Metric	Score
Accuracy	0.924545
F1 Score	0.925767
Precision	0.934850
Recall	0.924545

Table 3: Performances obtained by the Statue classifier.

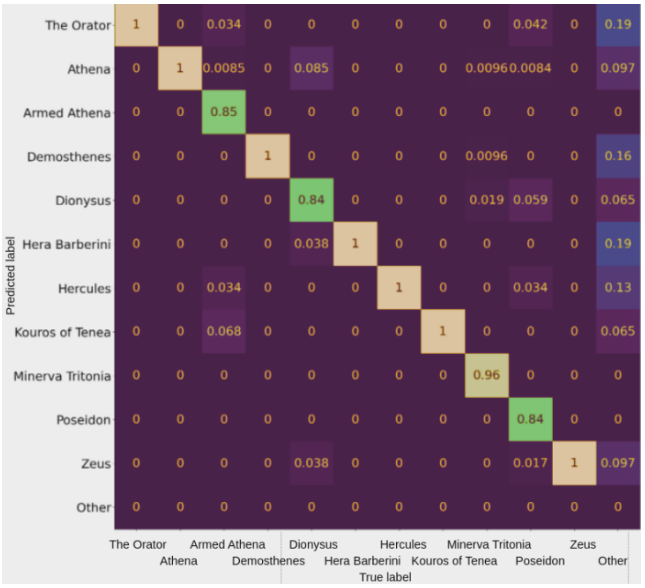

Figure 6: Confusion matrix obtained by the Statue classifier model evaluated on the Gypsoteca test set.

5.2 Live test in the Gipsoteca Museum

We set up an experiment in a real environment i.e. testing the functionalities of the application in the Gipsoteca Museum of the Sapienza University of Rome. Unfortunately, by the time of the live test 2 statues out of the 11 selected had been missing, so the test has been focused on the remaining 9 statues, recording 10 videos for each statue.

Moreover, the live test has been performed with different lights condition with respect to the one during the dataset creation; for this reason, some accuracy differences could be expected. In this scenario, the face detector successfully detected a statue face in 87 videos out of 90, with an accuracy of around 96.67%. From Table 4 it can be noticed that, once the face is detected, almost 100% of the statues are correctly classified, 87 out of 87 for the Human/Statue classification task and, 86 out of 87 for the Statue classification one. The whole pipeline obviously depends on the quality of the video recorded: the statues need to be recorded without abrupt changes in the framing of the subject, and out-of-focus videos need to be avoided in order to obtain a high-quality deepfake.

To rate the deepfake video quality we used a subjective evaluation using a scale of four values ("very good", "good", "quite good", "bad"), and the results reported in Table 4 are the average of the grades given by 10 visitors of the Gipsoteca museum chosen for the evaluation. In particular, there are some statues in which the produced deepfake is of high quality with a smooth transition among frames, like the statue of Hera Barberini (Figure 7 on the left) and Poseidon. We find out that those statues have a correct illumination that fosters the detection and the creation of the deepfake. Some others, instead, are subjected to excessive lighting that makes the subsequent process much harder. This is the case of the statues like Zeus and Kouros of Tenea (Figure 7 on the right).

We performed some additional tests on statues not present in our collected dataset, in order to consider the app behavior in the case of "outliers" statues; the results are shown on the last row of the Table 4. For this test, we used three main types of statues:

- Statues very similar to the ones of the dataset. The detector most of the time is able to say that the statue is "other", while in some cases they have been wrongly associated with a dataset statue.
- Statues exposed to an excessive amount of light or with half-hidden face: the face detector is fooled most of the time.
- Statues very different from the ones in the dataset: this was the easier case and all of them are correctly classified as "other".

In the end, considering all the results obtained, the pipeline works considerably well and the quality of the deepfake is considered pretty high by the users so overall the application seems a valuable tool to engage users in museums. Finally, we would like to remark that the pipeline of the three deep learning algorithms which are running on the server backbone requires 2 seconds of computational time for each second of generated deepfake video. This value can be affected by the employed hardware (Nvidia TitanX V Maxwell version in our case) and by further code optimization which will be performed in future works.

Figure 7: The statue of Hera Barberini (left) and Kouros of Tenea (right)

6 CONCLUSION AND FUTURE WORKS

In this paper, we presented an application that exploits an interactive techniques in the context of cultural heritage that leverage the possibilities to create and animate museum sculptures by creating deepfake videos. Any user with a smartphone may use the app to learn more about artworks at any museum that exhibits artistic artifacts with human forms and faces. A demo video of the system is available at the link on the GitHub platform [2] together with some produced deepfakes. The results demonstrated that the proposed pipeline can recognize the statue as well as the statue's face with high accuracy given that data associated with that statue is provided. In the future, our goal is to improve the deepfake generation process to reduce the time for the creation of the video and to achieve better qualitative results. In particular, to improve the accuracy, we want to upgrade the face detection part and solve the illumination problem for some statues in specific parts of the museum. Moreover, we foresee improving the response time of the server app in distributing the video. To better understand the goodness of the system a larger experimentation in the museum with more users would be necessary together with an increment of the number of statues/subjects censused in the dataset. To improve the evaluation phase and understand better and more easily the results of our project, we want to create a dataset based on people's votes for each deep fake video, and with it, train a classifier that return automatically the quality of the deep fake (based on the four categories given).

[2]https://github.com/Matteozara/Why_dont_you_speak.git

7 ACKNOWLEDGMENTS

This study has been partially supported by SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU and Sapienza University of Rome project EV2 (003 009 22). We thank Dr. Claudia Carlucci and Prof. Giorgio Piras, respectively the area curator and the director of the Classic Art Museum of Sapienza, for giving us the opportunity to collect and use the pictures and videos of the Gipsoteca statues to train and test the app algorithms.

Statue	N.Video	Face detection		H/S classifier		Statue classifier		Deep fake quality
		Detected	Not detected	Yes	No	Correct	Not correct	
Poseidon	10	10	0	10	0	10	0	very good
Hercules	10	9	1	9	0	9	0	good
Dionysus	10	10	0	10	0	10	0	good
Hera Barberini	10	10	0	10	0	10	0	very good
Zeus	10	10	0	10	0	10	0	good
Minerva Tritonia	10	9	1	9	0	8	1	very good
The Orator	10	10	0	10	0	10	0	very good
Demosthenes	10	10	0	10	0	10	0	very good
Kouros of Tenea	10	9	1	9	0	9	0	good
Total	90	87	3	87	0	86	1	
Other	15	12	3	12	0	7	5	good

Table 4: Results of the experiments done in Gipsoteca museum involving 10 random museum users.

REFERENCES

[1] 2014. *ImageNet site.* https://www.image-net.org/index.php

[2] 2018. *MediaRecorder Android.* https://developer.android.com/reference/android/media/MediaRecorder

[3] 2021. *Camera 2 Android.* https://developer.android.com/training/camera2

[4] 2021. *DeepFaceLab git repo.* Retrieved April 27, 2023 from https://github.com/iperov/DeepFaceLab

[5] 2022. *DeepFaceLive git repo.* Retrieved May 30, 2023 from https://github.com/iperov/DeepFaceLive

[6] 2023. *EfficientNet source code PyTorch.* https://pytorch.org/vision/main/_modules/torchvision/models/efficientnet.html

[7] 2023. *Faceswap git repo.* Retrieved July 12, 2023 from https://github.com/deepfakes/faceswap

[8] 2023. *Flask Python.* https://flask.palletsprojects.com/en/2.3.x/

[9] 2023. *HTTPURLConnection Android.* https://developer.android.com/reference/java/net/HttpURLConnection

[10] Zhuo Chen, Chaoyue Wang, Haimei Zhao, Bo Yuan, and Xiu Li. 2022. D2Animator: Dual Distillation of StyleGAN For High-Resolution Face Animation. In *Proceedings of the 30th ACM International Conference on Multimedia* (Lisboa, Portugal) *(MM '22)*. Association for Computing Machinery, New York, NY, USA, 1769–1778. https://doi.org/10.1145/3503161.3548002

[11] David Peter Fox. [n. d.]. https://play.google.com/store/apps/details?id=com.talkingstatues&hl=en&gl=US

[12] Ailin Huang, Zhewei Huang, and Shuchang Zhou. 2022. Perceptual Conversational Head Generation with Regularized Driver and Enhanced Renderer. In *Proceedings of the 30th ACM International Conference on Multimedia*. ACM. https://doi.org/10.1145/3503161.3551577

[13] Kaggle. 2020. *Dataset of human faces.* Retrieved 2020 from https://www.kaggle.com/datasets/ashwingupta3012/human-faces

[14] Imran Khan, Ana Melro, Ana Carla Amaro, and Lídia Oliveira. 2020. Systematic review on gamification and cultural heritage dissemination. *Journal of Digital Media & Interaction* 3, 8 (2020), 19–41.

[15] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. 2017. Fast Face-swap Using Convolutional Neural Networks. arXiv:1611.09577 [cs.CV]

[16] Yisroel Mirsky and Wenke Lee. 2021. The Creation and Detection of Deepfakes. *Comput. Surveys* 54, 1 (jan 2021), 1–41. https://doi.org/10.1145/3425780

[17] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. 2023. Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold. arXiv:2305.10973 [cs.CV]

[18] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, and Weiming Zhang. 2020. DeepFaceLab: A simple, flexible and extensible face swapping framework. *CoRR* abs/2005.05535 (2020). arXiv:2005.05535 https://arxiv.org/abs/2005.05535

[19] K. R. Prajwal, Rudrabha Mukhopadhyay, Vinay P. Namboodiri, and C. V. Jawahar. 2020. A Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild. *CoRR* abs/2008.10010 (2020). arXiv:2008.10010 https://arxiv.org/abs/2008.10010

[20] Albert Pumarola, Antonio Agudo, Aleix M. Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2018. GANimation: Anatomically-aware Facial Animation from a Single Image. arXiv:1807.09251 [cs.CV]

[21] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. 2019. MirrorGAN: Learning Text-to-image Generation by Redescription. arXiv:1903.05854 [cs.CL]

[22] Lawrence Rabiner and Ronald Schafer. 2010. *Theory and applications of digital speech processing.* Prentice Hall Press.

[23] Karen Robson, Kirk Plangger, Jan H. Kietzmann, Ian McCarthy, and Leyland Pitt. 2015. Is it all a game? Understanding the principles of gamification. *Business Horizons* 58, 4 (2015), 411–420. https://doi.org/10.1016/j.bushor.2015.03.006

[24] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-A-Video: Text-to-Video Generation without Text-Video Data. arXiv:2209.14792 [cs.CV]

[25] Ultralytics. 2023. *YOLOv8 official GitHub repository.* Retrieved July 27, 2023 from https://github.com/ultralytics/ultralytics

[26] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters* 23, 10 (2016), 1499–1503.