Shuttographer - Final Report

Shutter - A new and improved robot photographer

Eason Ding
Tetsu Kurumisawa
James Rosen
Bhavya Kasera

ACM Reference Format:
Eason Ding, Tetsu Kurumisawa, James Rosen, and Bhavya Kasera. 2025. Shuttographer - Final Report: Shutter - A new and improved robot photographer. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ABSTRACT

Shuttographer is an application-focused project that aims to make Shutter a better portrait photographer. Shuttographer employs a combination of computer vision, machine learning, and robotics control algorithms to identify a subject, orient towards the subject, capture an aesthetically pleasing photo, and edit the photo according to the subject's prompt. This system utilizes a state-of-the-art body-joint detector provided by Azure Kinect SDK [4] and geometric calculations to recognize and track a subject's face. To enhance the photographic capabilities, we integrate a portrait evaluation model trained on the PIQ23 dataset to pick the best out of a set of captured photographs. Shuttographer incorporates a Speech-to-Text model that allows the user to input a prompt, after which the system edits the best photo by putting the photo and prompt through a Stable Diffusion model. We conducted experiments in diverse settings that tested the overall performance of Shuttographer. Results indicate that Shuttographer is able to frame the person's face accurately but lacks techniques to take impressive portraits. However, by applying Stable Diffusion at the end of our pipeline, the resulting edited photos are aesthetically pleasing.

2 INTRODUCTION

Shutter is a robot photographer built by Yale's Interactive Machines Group [11], designed for applications in social robotics and human-robot interaction. The goal of the Shuttographer project is to make Shutter a better portrait photographer, both in terms of (i) the quality of the photographs it takes, and (ii) the interactions it has with its subjects.

With respect to (i), what makes a photograph 'good' or 'bad' is a question central to human perception and aesthetics, and touches on philosophical questions about the nature of beauty, emotion and meaning. For Shutter to become a good photographer, it must be able to understand and emulate human aesthetics, which is a challenging but fascinating problem [1]. If the problem is solved, then it will be able to interact with humans in a far more meaningful and engaging way, which is crucial for the development of social robotics. There are many techniques professional photographers use to shoot impressive portraits, such as centering the subject, focusing on the subject's eyes, and utilising indirect lighting. In order to replicate some of these techniques, we build a novel face tracking system, which uses geometric tracking alongside an imitation learning model to center a subject's face in the camera frame at an attractive angle. Although this by no means solves the problem of understanding human aesthetics, we hope that it is a step towards higher quality human-robot interactions. To further increase the quality of the resulting portrait, our system takes multiple photographs that are then evaluated by a convolutional neural network we call the portrait evaluation model, and the photograph with the highest quality score is selected.

With respect to (ii), it is clear that the strength of a photographer does not rest solely in the qualities of the photographs they take, but also in the quality of the interactions they have with their subjects. Thus, a significant part of making Shutter a better portrait photographer involves improving its interactions with participants. In particular, we hold that it is very important that participants have a fun time during their interaction with Shutter and that they are at ease throughout the process. In order to improve this aspect of Shutter's photography, we allow subjects to edit the photographs Shutter takes with Stable Diffusion, based on a prompt that they provide. This gives participants some agency over the creative process, which leads to higher quality interactions with Shutter.

There are several possible applications of the Shuttographer project. The recent development of Amazon's Astro, a home robot assistant, shows the potential commercial impact of a robot that can act as a general-purpose assistant. If a robot like Astro could capture family moments in a beautiful and meaningful way, this would significantly add to the value of a robot assistant. There are also applications in the domain of accessibility: if, for example, someone with a physical disability is unable to take photographs, a photographer like Shutter that is capable of taking impressive photos in a fun and engaging way could add a lot of value to that person's life, allowing them to capture moments they otherwise could not.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 RELATED WORK

Social robotics is a subfield of human-robot interaction (HRI), where the goal is to develop robots that interact with humans in a meaningful, context-aware way that is sensitive to social cues and norms [10]. Shutter is a robot photographer that was designed for public, in the wild human-robot interactions [11]. It was built with the purpose of studying fundamental questions in HRI, such as [12]:

- How should a robot initiate social interactions.
- How can a robot adapt to the preferences of its users.
- What does it mean for a robot to take a good photograph.

Lots of exciting work has been done with Shutter in recent years, including a paper that turned the robot into a 'humorous' photographer by having it display jokes to users before taking their photographs [13].

Work on learning to take good photographs with a robot photographer includes [14], which shares an important element of our work, in that it uses a convolutional neural network to evaluate photographs. However, the robot used is not fully interactive: the robot does not truly 'learn' to take good photographs, but instead takes multiple photos at different angles and then evaluates their quality post-facto. By contrast, in our project, we teach Shutter to take good photographs by tracking the face and fine-tuning the framing using the imitation learning model.

Other work in this area includes [22], which integrates the robot with a smart phone to optimize group photographs. The paper in [23] presents a mobile robot photographer, which uses facial detection to find subjects and approaches them to take a photograph. The robot frames the subject based on the detected centre of the face, rotating the camera and adjusting its frame until the centre of the subject's face is in the centre of the image. In contrast, the combined approach using facial tracking and imitation learning in our method allows for finer adjustment of the framing, and allows for a variety of poses in the captured photograph. Using Stable Diffusion on the photograph selected by the Portrait Evaluation Model, our system also allows customization of the portrait based on the user's preferences. Hence, we use a unique blend of automated portrait photography and evaluation with human input to produce the final portrait.

4 METHOD

Our approach involves several components that work together to make Shutter a better portrait photographer.

4.1 Updates to initial plan

Initially in the project proposal, we had planned to use a segmented version of Shutter's camera input as the state space for the imitation learning portion. In particular, we were going to use the OpenCV library Cascade classifier [9] to draw boxes around human faces that come into view of Shutter's camera, and use the 2D coordinates of the four corners as the state space. However, upon attempting to run the OpenCV library code, we found that there were three problems with this approach:

(1) The approach provides a 2D coordinate, which does not take into account the depth (distance of the subject from the camera)
(2) The approach requires more computational power and time, since we have to run the classifier for each frame of the input video stream at run time.
(3) The approach only considers the face and not the whole body of the subject.

Thus, we decided to use the Kinect camera and depth camera in order to detect where our subject is in the 3D environment. We analyzed the Kinect depth camera, and we performed human body segmentation and body tracking from Azure Kinect Body Tracking SDK on the input depth image.

Additionally, in our milestone report, the planned pipeline for the project was to frame the subject within Shutter's camera by using a policy learned through behavior cloning. However, upon attempting to train Shutter to frame the subject, we realized that this phase could be substituted by a target tracking algorithm instead. Thus, we decided to create a face tracking model, which tracks the subject's face (as measured through Kinect's body tracker).

Furthermore, we decided to substitute the Reinforcement Learning (RL) phase with behavior cloning. There were two reasons for this decision:

(1) RL would require a lot of safety oversight and measures to be taken so that Shutter did not exceed its limits in movement and physical capabilities.
(2) The benefit of RL, which was to capture the policy that would allow Shutter to take a "good" portrait photo, could be substituted by behavior cloning, in which humans show demonstrations for how to take a good portrait photo and Shutter learns the policy by imitating them.

Since the RL stream was substituted by the imitation learning component, the portrait evaluation model, which we initially planned to use as a reward function in the RL component was decided to be used to pick the best photos out of the 10 photos that Shutter takes.

Finally, we decided to add several features to our project that made participants' interactions with Shutter more interactive and fun, embracing Shutter's role as a social robot. This included having Shutter ask participants for consent before taking their photo, as well asking participants to provide a prompt that we used to edit their photos with Stable Diffusion.

4.2 General Project Pipeline

In the first stage of our pipeline, Shutter is inactive until a person comes into view of the Kinect camera, which is detected using the Kinect's in-built body tracking features. This triggers the Face Tracking system (discussed below in Section 4.3 and 4.4), and also triggers a ROS node to play an audio recording which says, "Hello, I'm Shutter, the friendly robot photographer! Can I take your photograph?". We then record the next 5 seconds of audio using ROS' audio capture package, and place this audio through a speech-to-text model (specifically, OpenAI's whisper-base model [19]), which transcribes the recorded audio into text. We wrap this text in the

following prompt: "I have just asked someone if they want their photograph taken, and they responded with {text}. Did they consent to having their photograph taken? Please only respond with 'yes' or 'no' (lowercase and no punctuation)". This prompt is then passed to GPT-3.5-turbo via OpenAI's API [20], which determined whether the participant has consented to having their photograph taken. This allows participants to respond to Shutter in a natural way, increasing the depth and quality of social interactions with the robot.

If the participant does not consent to having their photograph taken, then an audio recording is played which says "That's a shame - let me know if you change your mind". Otherwise, another audio recording is played that asks users for a prompt that will be used to edit their photo. After recording audio for 10 seconds and processing it with our speech-to-text model, Shutter says "Sure thing, coming right up", and begins to 'take photos' of the participant. In reality, we simply sample 10 frames over a 5 second period, and then pass these frames through our portrait evaluation model (discussed below in Section 4.5). We take the photo that is given the highest score by the portrait evaluation model, and then pass this image to a Stable Diffusion API [21] with the prompt provided by the participant. Once this is completed, an audio recording is played which says "Your photos are all done, I can't wait to show you!". We then manually show the participants their best photo and their edited photo.

4.3 Face Tracking

The face tracking system in our approach is to make Shutter frame a person for a portrait using geometry according to the person's nose location. This approach is inspired by [18], and we worked on top of the code by combining the face tracking process into a simple finite states machine, where we defined the rest state (when no person is detected), tracking state (when someone enters the frame), and fine-tuning state (when the person stays still and ready to be filmed).

The action for rest state is to return to Shutter's rest position with 4 joint angles to be

$$[0, -1.5, -1.5, 0]$$

The action for tracking state is to let Shutter looking towards the person's nose with 4 joint angles to be

$$[\textbf{joint } \mathbf{1}_G, -1.5, \textbf{joint } \mathbf{3}_G, 0]$$

Where **joint** $\mathbf{x}_G$ means the result from Geometry calculation for joint angle $\mathbf{x}$. Also, we use memory mechanism such that Shutter remembers the person it is tracking and will keep looking at the same person if multiple people are inside the frame.

The action for fine-tuning state is to fine-tune Shutter's position that better films the person. The goal position of Shutter's 4 joint angles are produced by imitation learning discussed below.

4.4 Imitation Learning

The imitation learning component's goal is to make subtle corrections after the face tracking algorithm frames the person to capture movements that a human would make in order to take a good portrait photo. The two upsides of imitation learning over the simple face tracking algorithm are: 1. It allows Shutter to move in a more natural trajectory when framing the person, which may affect how the subject poses in a portrait photo 2. People may not necessarily try to center the photos around the face depending on their pose when taking a portrait photo.

4.4.1 Data Collection & Dataset: The data was collected through demonstration by a human using a teleoperation joystick. We positioned a subject in front of Shutter, and the demonstrator operated the joystick to move Shutter's position to frame the subject in a position in the frame that they felt was a good position for a portrait photo. At the same time, we recorded the head position of the subject measured through the Kinect body tracker and the joint positions. This yielded 20 minutes worth of data that we could use to train the model. In terms of data processing, since the demonstrator would start moving Shutter when the subject moved into a new position, and stop moving it when the position seemed suitable for a photo to be taken, we parsed out the data to when there was continuous input from the joystick. Then, we constructed a state action pair, which the model was trained upon. This was done by taking the 3D coordinate of the face provided by Kinect, which we then transformed into the 3D coordinate relative to the Shutter base_link, and joint positions (for joint1, 2, 3, and 4), at time t as the state, and the action as the joint positions at time t+1. The time was calculated through the second that is published from the Shutter's internal clock.

4.4.2 Method: Once we obtained the state action pair, we trained a simple neural network model. The model was a feedforward neural network with one input layer, two hidden layers, and one output layer. The model transforms the state (7 features) at a certain time step through the layers using rectified linear unit (ReLU) activation functions and produces a final output with 4 units, which were the predicted joint positions in the next time step.

Figure 1 visualizes the Kinect body tracker. The lower-left image is the segmented human body, and the center image is the predicted body joints. The joystick control during the imitation learning data collection is shown in Figure 2.

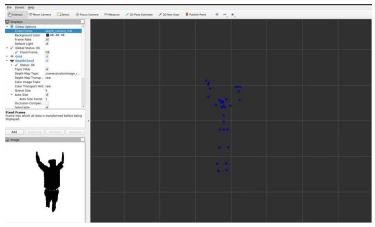

Figure 1: Human body segmentation and body tracking

4.5 Portrait Evaluation Model

Out of the various photographs taken by Shutter, we must select the best one. To do this, we train a convolutional neural network, which we call the Portrait Evaluation Model, on the PIQ23 dataset [5]. The model outputs probabilities that the photo belongs to each

Figure 2: Shutter joystick control by Tetsu

of the 9 quality classes, ranging from score 0 to score 8, and the class with the highest probability is picked as the quality score for the input image. The photographs taken by Shutter are evaluated using this model and the one with the highest quality score is selected as the best photograph and used in the next part of the pipeline, to customize the photo using Stable Diffusion.

4.5.1 Dataset: We trained our model on the PIQ23 dataset, which contains 5116 smartphone portraits taken by people in a variety of contexts, organised by setting (indoor, outdoor, night scene, lowlight), with quality scores for each portrait. The scores cover various metrics such as exposure, details, etc, but we used the overall quality score included in the dataset to train the model, which ranges from 0 to 8. Figure 3 shows a few examples from the dataset.
As a preprocessing step, we filtered out the indoor portraits in the dataset, with their respective overall quality scores as the labels, as this data would be the most relevant for our project. This gave us a dataset of 1397 images, which we then split into training, validation, and test sets using a 8:1:1 split. We also resized all the images to be 224x224 to feed into the model.

4.5.2 Model: The model architecture we used for training was the ResNet18 model [15] with pretrained ImageNet weights, available through the torchvision library [16]. An overview of the model architecture can be found in Figure 4. The model has 5 convolutional layers, each followed by a 3x3 max pool, and a 7x7 average pool layer in the end. The last layer, which is the fully connected layer, was modified in our model to get a 9-dimensional output vector, containing the probabilities for each of the classes. We finetuned all the weights in the model by training on the PIQ23 dataset.

5 EXPERIMENTS

5.1 General Project Pipeline

We tested our full pipeline with a variety of different natural language responses to Shutter's consent request and prompt request. A representative example of the final output of Shuttographer for the prompt "A wizard against a starry background", including the best photo selected by our portrait evaluation model and the edited

Figure 3: Examples from the PIQ23 dataset with quality score 0 (top left), quality score 2 (top right), quality score 7 (bottom)

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 4: ResNet18 model architecture [17]

photo produced by Stable Diffusion, is presented in Figure 5. In order to qualitatively evaluate the performance of our overall pipeline, we had a group of 9 friends participate in a survey, in which they watched a video of Shutter taking and editing photos and were asked to assess the robot against a number of different metrics. The survey can be found here. The survey indicates that, overall, participants perceived Shutter as a friendly robot, and that they would like to have their photos taken by Shuttographer (see Figure 6). However, participants did not think that Shutter's movements were particularly smooth or lifelike (see top of Figure 7), and participants

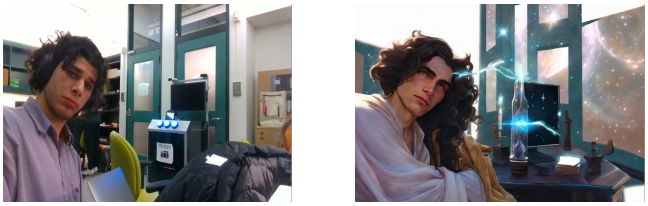

Figure 5: The best photo chosen by our portrait evaluation model (left), and the corresponding photo edited with Stable Diffusion (right), with prompt "a wizard against a starry background"

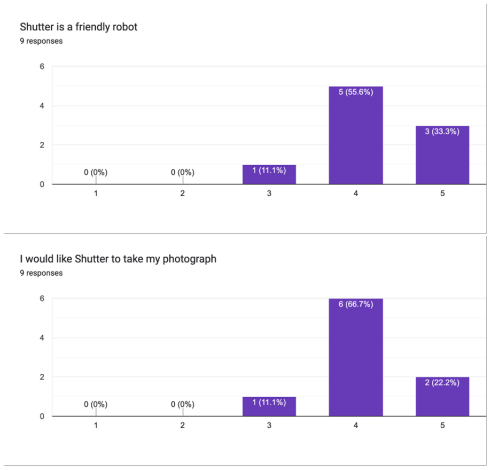

Figure 6: Survey responses for overall impression of Shuttographer

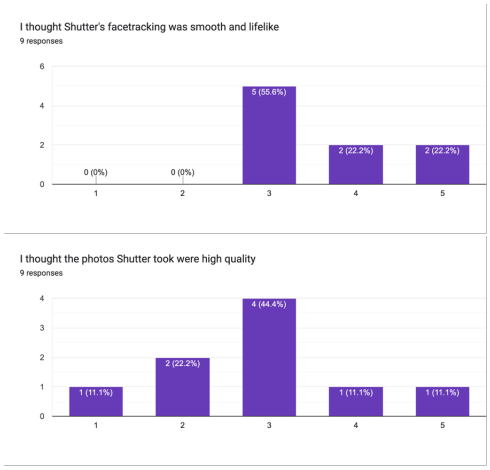

Figure 7: Survey responses for the quality of Shutter's movements (top), and the quality of Shutter's photos (bottom)

were even less impressed with the quality of the photos taken by Shutter (see bottom of Figure 7).

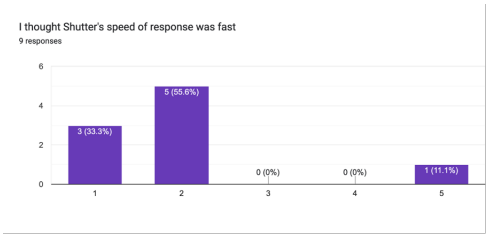

Figure 8: Survey responses for question about the latency of Shuttographer

5.1.1 Natural Language Processing: Although we did not perform a quantitative evaluation of the speech-to-text model for processing participant's consent responses and prompt responses, we found that the model correctly transcribed speech almost universally, even in instances with low-to-medium levels of background noise. The model would sometimes fail to transcribe speech when the participant spoke very fast, or when there were high levels of background noise. However, even in these cases, often enough semantic information was retained from speech that GPT-3.5-turbo was able to correctly determine whether consent had been given.

In order to evaluate GPT-3.5-turbo's performance at determining whether participants had provided consent to having their photo taken, we tested the model on five different affirmative responses to the consent request: 'Yeah sure', 'No problem', 'Go ahead', 'Feel free', and 'Go for it', as well as four different negative responses to the consent request: 'No thanks', 'I'm ok', 'I'd rather not' and 'Please don't'. The model returned the correct response with 100% accuracy.

One of the metrics we asked participants about in our survey was the latency of the system, which primarily occurred during natural language processing. Across 10 trials, the average amount of time it took for Shutter to respond to a participant after it had finished recording was 3.6 seconds. However, often the audio recording would continue for long after the participant had finished speaking (as we recorded for a fixed number of seconds), which often added to latency. In the survey, participants felt that latency was a significant problem with Shuttographer (see Figure 8).

5.2 Imitation Learning

Model Performance

The imitation learning model, which was a 4 layer deep neural network, was trained on a dataset that included over 500 state action pairs, and the validation loss was 0.0176 and the validation MAE was 0.0783 in radians of the joint position, and the plots can be seen in Figure 9. Due to the small number training data, the model started over-fitting as seen in Figure 10, and thus the accuracy we achieved was the best we could achieve. However, as mentioned in the next section, when we evaluate the performance of the imitation learning model on the physical Shutter robot, it mimics the human demonstration well.

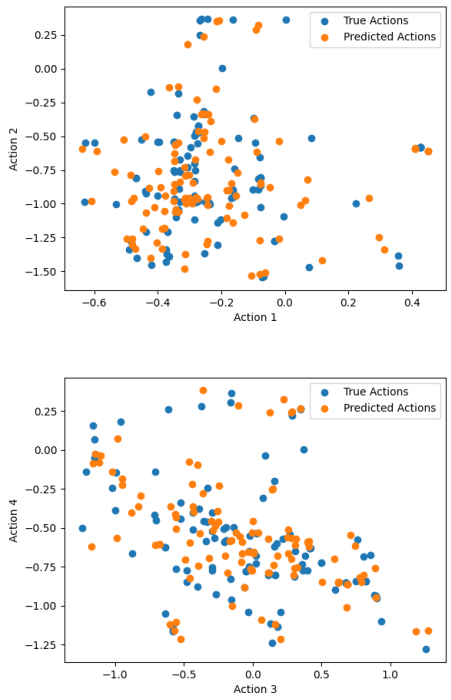

Figure 9: Plots of joint positions predicted by model vs true joint position for joint 1, 2 (top) and 3, 4 (bottom)

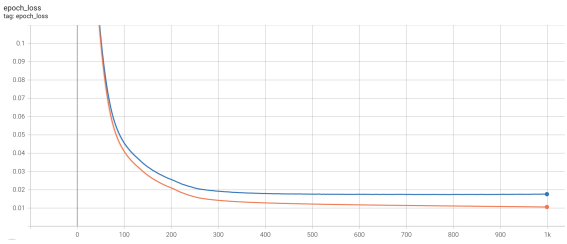

Figure 10: Model Loss (Blue: Validation Loss, Orange: Training Loss)

5.3 Face Tracking

We tested each state of Face Tracking, the result is manually evaluated to determine if Shutter makes the desired movement.

5.3.1 Rest State: We tested this state by creating an environment where no one can be detected by Kinect camera, and then we observe Shutter's behavior. We tested Shutter's behavior under the following two situations:

(1) No one is inside the Kinect camera frame
(2) A person moves out of the Kinect camera frame

In all situations, Shutter successfully returns to its resting position.

5.3.2 Tracking State: One person is asked to move randomly at different positions in front of Shutter, and we looked into the outputs from realsense camera to determine if the person's face is at the center of each photo. We tested Shutter's ability to face towards the person at four relative positions: left, front, right, and top. In conclusion, Shutter successfully frames the person at different positions and locates the face at the center of each photo while the person is moving.

5.3.3 Fine-tuning State: One person is asked to stand still at different positions in front of Shutter, and we looked into the outputs from realsense camera to determine if Shutter adjusts its camera to take a better photo. We tested Shutter's ability to film the person at four relative positions: left, front, right, and top. In conclusion, Shutter is good at capturing the person at front, but it fails to capture the person at other locations, especially when the person is standing at the left or right of Shutter. Since we are doing behavior cloning, one possible explanation is that we failed to collect enough data that covers all the states.

5.3.4 Trials to combine Imitation Learning and Geometry Tracking: Since the goal of Imitation Learning is to fine-tune the result from Geometry Tracking, we tried to combine the result from both by updating Shutter's joint positions as:

$$[\textbf{joint } \mathbf{1}_G, -1.5, \textbf{joint } \mathbf{3}_G, \textbf{joint } \mathbf{4}_I]$$

Where **joint** $\mathbf{x}_I$ means the result from Imitation Learning for joint angle $\mathbf{x}$. We let the Imitation Learning to take over only Joint 4 since Joint 4 controls the orientation of Shutter's camera and doesn't affect the overall Shutter's joint positions, and we performed the same experiments as we tested the Fine-tuning State. In conclusion, Shutter's behavior is desirable and it can always film the person at the center of the photo. Further, Shutter tries to elevate its camera to the same height as the person's head, which is a great improvement from face- tracking by Geometry calculation. However, when we tried to run all our nodes (including Portrait Evaluation Model and Photo Editing nodes), we ran out of GPU usage. Thus, we had to abandon the fine-tuning State for our project since the Tracking State already produces good enough photos.

5.4 Portrait Evaluation Model

To test the Portrait Evaluation Model, we calculated its accuracy on a held-out test set from the PIQ23 dataset. The model achieved an accuracy of 31.8% on the test set consisting of 280 images, i.e. it classified the image as the correct quality score (classes 0 - 8) in 31.8% of the cases. Since image quality is a subjective metric, we also calculated the accuracy of the model in a window of $+/-1$ around the ground truth quality score, and in this case the model achieved an accuracy of 45%.

Figure 11 shows the distribution of the error (true score - predicted score) over all the test images. In general, the score predicted by the model seems to be lower than the true score, with a mean error of 1.66. The mean absolute error over the test set was 1.92.

Figure 12 includes a few examples of the model's performance on the test set, showing misclassified and correctly classified images with different quality scores.

The low accuracy of the model on the test samples is likely due

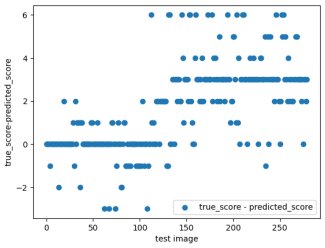

Figure 11: Distribution of (true score - predicted score) for the Portrait Evaluation Model over the test set [17]

Figure 12: Examples of the Portrait Evaluation Model performance on the test set - predicted score 0, true score 7 : misclassified (top left), predicted score 0, true score 0 : correctly classified (top right), predicted score 4, true score 1 : misclassified (bottom left), predicted score 6, true score 6 : correctly classified (bottom right)

to high noise in the true quality scores, as well as the highly subjective nature of the quality scores, which would be difficult for a model to learn since they cannot be directly correlated with factors such as brightness, contrast, sharpness etc. However, with a mean absolute error of 1.92, our model can still be used as a fairly reliable indication of whether a portrait is 'good' or 'bad'.

Figure 13 shows a set of pictures taken by Shuttographer that were evaluated by the model, and the picture with the highest predicted quality score.

5.5 Ros Nodes and Launch file

Finally, we tested our pipline by running all the components simultaneously. However, we encountered two problems when running our nodes, Shutter, Kinect camera, and realsense camera altogether. Below are the solutions to each of the problems we encountered.

Figure 13: Photos taken by Shuttographer that were input into the Portrait Evaluation Model, and the best photo picked by the model (bottom right)

5.5.1 *Problem: when Shutter and Kinect camera are launched at the same time, Shutter's tf tree would disappear.*

Solution: This problem is caused by the launch file of Kinect camera overwriting robot description. In this case, both of the launch files create a node called "robot_state_publisher", and Shutter's node of the same name will automatically stop running. We solved this problem by passing an argument called "overwrite_robot_description:=false" to Kinect launch file.

5.5.2 *Problem: when Kinect camera and realsense camera are launched at the same time, realsense camera cannot output any image.*

Solution: This problem is caused by the USB hub we used for transfer messages. To solve this problem, we need to use different USB portal for each camera to avoid interference.

5.5.3 *Launch file.*

By encapsulating the configuration of multiple nodes into a single launch file, users can more easily manage and reproduce our experiment results. In the launch file, we run the following processes in sequence: realsense camera rs_camera.launch, Kinect camera driver.launch, Shutter shutter_with_face.launch (with one modification of the line: <node name="controller_spawner" pkg="controller _manager" type="spawner" respawn="false" output="screen" args= "joint_group _controller "/>. We removed the '–stopped' flag to activate joint_group _controller at the start), static transform from Kinect to Shutter, face tracking node, asking for user consent node, audio recording node, prompt recording node, audio capture node, photo editing node, and new person detector node.

The goal of this launch file is to simplify the process of starting and configuring multiple ROS nodes simultaneously. In our experiments, the launch file starts all the nodes as we desired.

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

Our goal with this project was to make Shutter a better photographer by increasing the quality of photographs it takes as well as improving its interactions with its subjects. To this end, our system performed well at framing the subject for a portrait photograph and at interacting with the subject. However, there were some limitations in our project that affected the quality of the photos taken by Shutter, as well as the photos selected by our portrait evaluation model. This was captured by the qualitative feedback from our survey, where we received relatively low scores in these areas. This reflects the difficulty of evaluating the quality of portraits, as well as the difficulties we encountered in integrating our imitation learning model into the overall pipeline. However, participants seemed to find interacting with Shutter very fun, and were excited to have Shutter take their photograph in person. This suggests that the project was a success from the perspective of social robotics and human-robot interaction.

6.2 Future Work

6.2.1 Imitation Learning: Although we could not include the imitation learning model into the demo due to the lack of GPU to process the model, a computer with increased computational capacity should be able to handle it. For the overfitting that we encountered when training the model, we suspect that this was due to the limited variety of demonstrations that the dataset was composed of. By increasing the demonstration, we should be able to avoid this problem. Additionally, in terms of encapsulating different styles of portrait photos, we believe that the imitation learning model holds a fascinating potential. Using the pipeline introduced in this project, Shutter could learn different ways to frame portrait photos from different human demonstrators, and take portrait photos that have different photographic styles.

6.2.2 Portrait Evaluation Model: One of the challenges we faced was the erratic performance of the portrait evaluation model due to the differences in the dataset on which it was trained and the real-world setting in which Shutter took portraits of people. To address this, finetuning the model by training it on photographs taken by Shutter that are evaluated by the subject each time Shutter takes a portrait photo could improve the performance of the model significantly.

6.2.3 Natural Language Processing: One of the most significant limitations of Shuttographer was the latency introduced by the NLP component of the project. A big aspect of this problem was that Shutter recorded responses of participants for a fixed length of time, and in order to ensure participants responses were never cut off before they had finished speaking, we recorded responses for an overly long period of time. A natural solution here that would lower latency would be to use a more sophisticated speech-to-text model that can detect when a participant has finished speaking. Furthermore, many of the responses to the consent question can be predicted in advance (the majority of people simply answer "yes"). If we check whether the participant's consent response matches some expected phrase before calling the GPT-3.5-turbo API, then we could avoid the API call in many cases and lower latency.

6.2.4 Overall: Our project is based on the assumption that only one person is interacting with Shutter. Although the presence of other people will not affect Shutter's ability to frame the target, Shutter needs to be able to respond to multiple people at the same time in the real-world setting. One potential solution is to design a graph neural network that predict the importance of each individual around Shutter, and future works could focus on improving one-to-many interaction between Shutter and users.

7 SUPPLEMENTARY MATERIAL

7.1 Code

https://github.com/EasonDi/shuttographer

7.2 Documentation

https://github.com/EasonDi/shuttographer/blob/main/shuttographer/README.md

7.3 Demo Video

https://drive.google.com/file/d/1CPIf0Y4r4z-jDVN1dsIFx7Lc3A_riCAc/view?usp=sharing

REFERENCES

[1] Jacko, J. A. (Ed.). (2012). Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications

[2] Haimes, P. (2021, May). Beyond Beauty: Towards a Deeper Understanding of Aesthetics in HCI. In Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-7).

[3] Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. ACM Computing Surveys (CSUR), 50(2), 1-35

[4] https://learn.microsoft.com/en-us/azure/kinect-dk/body-joints

[5] Chahine, N., Calarasanu, S., Garcia-Civiero, D., Cayla, T., Ferradans, S., & Ponce, J. (2023). An Image Quality Assessment Dataset for Portraits. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 9968-9978).

[6] https://github.com/ultralytics/ultralytics

[7] https://github.com/DXOMARK-Research/PIQ2023

[8] https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html

[9] https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html

[10] Sheridan, T. B. (2016). Human–Robot Interaction: Status and Challenges. Human Factors, 58(4), 525-532. https://doi.org/10.1177/0018720816644364

[11] Lew, Alexander, Sydney Thompson, Nathan Tsoi, and Marynel Vázquez. "Shutter, the Robot Photographer: Leveraging Behavior Trees for Public, In-the-Wild Human-Robot Interactions." arXiv preprint arXiv:2302.00191 (2023).

[12] https://interactive-machines.gitlab.io/projects/photographer.html

[13] Adamson, T., Lyng-Olsen, C.B., Umstattd, K., and Vázquez, M. (2020). Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI).

[14] Newbury, Rhys, et al. "Learning to take good pictures of people with a robot photographer." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

[15] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[16] https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html

[17] https://www.researchgate.net/figure/ResNet-18-Architecture_tbl1_322476121

[18] https://github.com/Yale-BIM/f23-assignments/blob/master/assignment-2/shutter_behavior_cloning/src/expert_opt.py

[19] https://huggingface.co/openai/whisper-base

[20] https://platform.openai.com/docs/models

[21] https://stability.ai/

[22] R. C. Luo, W. U. Chan and P. -J. Lai, "Intelligent robot photographer: Help people taking pictures using their own camera," 2014 IEEE/SICE International Symposium on System Integration, Tokyo, Japan, 2014, pp. 322-327, doi: 10.1109/SII.2014.7028058.

[23] S. Suzuki and Y. Mitsukura, "Mobile Robot Photographer," 2013 2nd IAPR Asian Conference on Pattern Recognition, Naha, Japan, 2013, pp. 742-743, doi: 10.1109/ACPR.2013.192.