# Lights! Camera! (Optimal) Action! Learning a Lighting Policy for Robot Photography

Joe Connolly
Department of Computer Science
Yale University
New Haven, Connecticut
joe.connolly@yale.edu

Kayleigh Bishop
Department of Computer Science
Yale University
New Haven, Connecticut
kayleigh.bishop@yale.edu

Simon Mendelsohn
Department of Computer Science
Yale University
New Haven, Connecticut
simon.mendelsohn@yale.edu

**Figure 1: Our photography lighting setup, including the 4-DoF Shutter robot and two lamps with remote-controllable Philips Hue Bulbs**

## ABSTRACT

Manually ensuring proper lighting when capturing portrait photos can be a time-consuming and tedious task, as well as difficult for novice users. In order to ease the burden of photographers, and also enable automated systems to take high quality photos, we designed a routine that automatically adjusts lighting to optimally prepare a robot photography system for capturing portrait photographs. We combine multiple different existing image quality metrics and use those metrics as a reward for a neural network system that learns how to optimize lighting. In our arrangement, the neural network system is connected to two Hue light bulbs whose brightness it can adjust however it sees fit. We achieve promising results and demonstrate the feasibility and practicality of such a robot photography system. We also highlight exciting avenues for future research.

## KEYWORDS

Reinforcement Learning, Supervised Learning, Q-Learning, Photography, Convolutional Neural Networks, Robotics

## 1 INTRODUCTION

In portrait photography, proper lighting of a scene is vital to capturing high-quality photos. Expert studio photographers must account for numerous variable factors including the equipment, the poses of both photographer and subjects, and qualities of the subject themselves. Through training and experience, they learn how to control their studio lighting to capture an aesthetically pleasing photograph that captures the image of the subject as desired.

Now more than ever before, photography has gone from an art dominated by experts to a daily activity for millions of people. The presence of digital cameras on smartphones and computers and the proliferation of these devices into households has meant that the ability to snap photos at a moment's notice has become a given for the everyday user. Concurrently, autonomous robot photographers are being developed for a number of applications, from photographing events like weddings to taking portraits [9, 14]. Despite these innovative hardware designs and computational methods, high-quality portrait photos remain nearly impossible to create without studio equipment and training of professional photographers.

The task of replicating professional-quality work has become a foundational problem for computational photography. The presence of "portrait mode," which artificially applies blur to the background of an image, is a selling point for popular smartphone brands like the iPhone [7]. Commonly available filters and image masks can mimic post-processing modifications popular in photography by modifying the exposure, white balance, brightness, and contrast of photographs using a variety of computational techniques [2]. However, these techniques still struggle to create the lighting effects of real studio lighting handled by professionals.

In this project, we seek to bridge this gap by designing a system that can learn to control remote-controllable lights to achieve aesthetically pleasing portrait photos. Our goal is to create a system that can help turn relatively structured environments into professional studios by automatically adjusting the connected lights to capture the best possible portrait.

Though portrait quality is subjective to a degree, there exists an agreed-upon standard of high-quality photography among both professionals and the general public. As such, there also exist computational methods that perform image quality assessment (IQA) and predict professional opinions with impressive reliability, ranging from simple metrics like image entropy to those using the power of CNNs to extract image features [17].

Building on recent work in image quality assessment, we leverage the power of deep reinforcement learning to design a system that can learn to optimize image quality as measured by these metrics. Our system is designed for the Shutter Robot. The system automatically gathers training data from the environment, including image data, lighting data, face poses, and the robot pose. This data is then input into a neural network, which through training learns to predict the best lighting action for the scene, given the current state of the world.

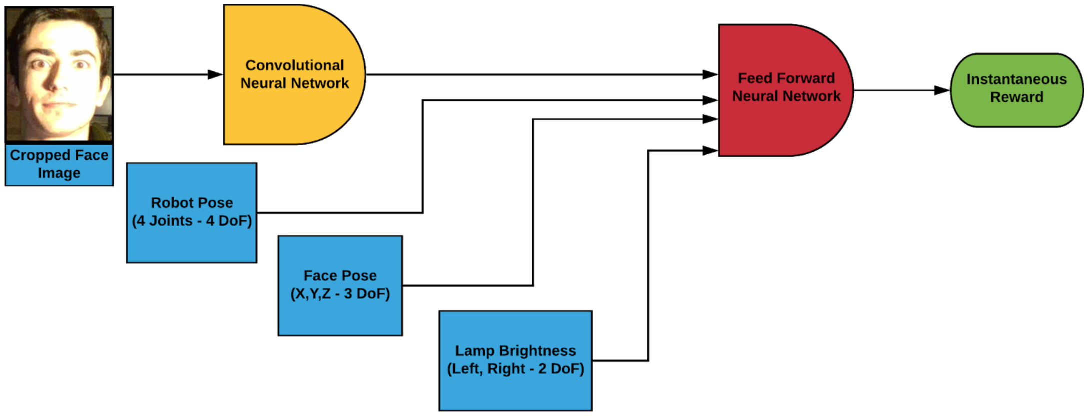


**Figure 2: The neural network architecture we leverage to estimate the instantaneous reward for reinforcement learning**

# 2 RELATED WORK

## 2.1 Image Quality Optimization/Assessment

Some previous works have explored the potential of deep CNNs to optimize the quality of photographs. Google's Creatism [4] is a deep-learning system designed for photography by learning not only how to best post-process an image, but also how to best compose (i.e. frame) a photo. The system includes its own deep image quality assessment network (or "scorer"), which learns separate aspects of aesthetics, including lighting, color composition, and framing. Using the trained scorer network and otherwise unlabeled professional quality photos as input, the network is then trained to optimize image quality given a particular image operation, such as saturation and cropping. A separate, lighting-related operation the authors call dramatic masking is trained using a Generative-Adversarial Network (GAN). The results from this system were then evaluated by professional photographers, with impressive results - 41.4% of the photos predicted by the network to be at or above semi-pro level were agreed to be at such a level by professionals. This system design and its results are exciting; however, the system is still learning to optimize a particular scene post-capture, after it chooses a scene to capture. The system does not change anything about the capture environment, as we aim to do.

Interestingly, the lighting optimization problem we examine is tightly coupled to similar problems in robotics. The Adaptive Lighting for NLOS system [3] optimizes the lighting position to maximize the radiosity of images beyond the line of sight of the camera attached to a robotic arm. This optimization is performed using the Sequential Least Squares algorithm over the discrete set of patches onto which the light might be shined. While the authors saw much success from this algorithm, their problem was limited in domain as they were optimizing for radiosity over all the patches - radiosity per patch could be measured relatively simply by lighting one patch at a time.

Perhaps the prior research that is most relevant to our specific problem formulation involves systems grappling with how to perform No-Reference Image Quality Assessment (NR-IQA) [1]. NR-IQA refers to the process of judging image quality with no ground truth images on which to base quality judgements. NR-IQA is particularly relevant to this problem context because collecting and hand-labeling data for referenced Image Quality Assessment would be extremely costly and time-consuming. That said, there are many famous NR-IQA architectures that have been published fairly recently and perform quite well on established datasets, including BRISQUE [12], BIQI [15], Hallucinated-IQA [11], and DIQA [8]. However, the challenge that these metrics present for our system is that they tend to measure spatial components of images to judge quality (e.g. blurring) more so than characteristics like brightness. However, some no-reference metrics such as image entropy [16] do show potential to optimize lighting due to their measurement of aspects such as image contrast and sharpness. Consequently, we believe these metrics can help form a solid basis for our system's instantaneous reward function needed for reinforcement learning.

## 2.2 Reinforcement Learning

Finally, work in the field of reinforcement learning, such as Google's DQN [13] and DDPG [10] has revealed the vast potential of deep reinforcement learning methods that use exploration. To achieve our goal, it makes the most sense to capture data in batches rather than perform explorative learning, given that lighting is a spectrum and can be explored relatively easily through simple discretized linear combinations of brightness. Unlike the sequential environments in DQN where Q-values are used as a measurement of eventual reward, our problem domain comes with simple one-step rewards thanks to our image quality metrics.

One consideration in this decision to collect data in advance was that standard off-policy deep reinforcement learning algorithms can suffer from extrapolation errors when training data differs considerably from the distribution under the current policy [5]. As a result, we made an effort to include a wide variety of training data that would closely resemble the policy distribution.

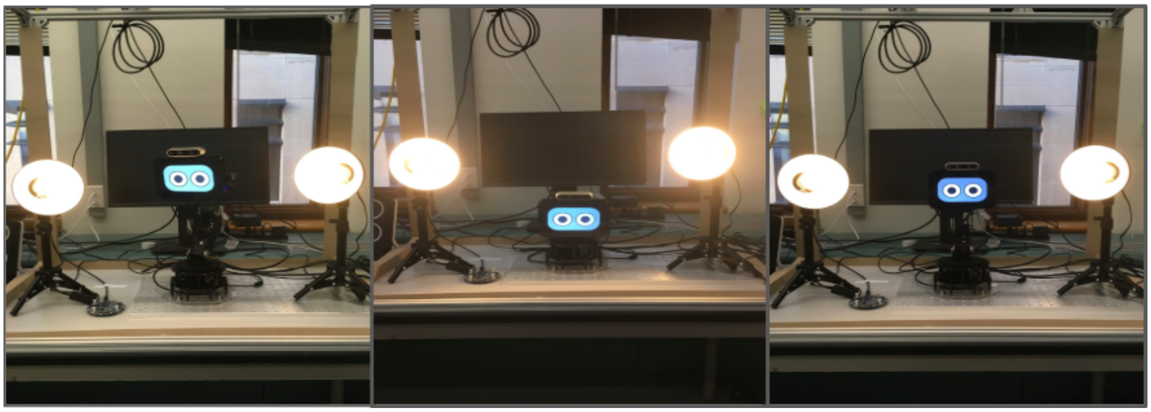

**Figure 3: Some of the different robot poses we collected data from for training our neural network. We varied both height and orientation for each pose, and at each pose we adjusted all the other parameters to ensure robust data collection.**

# 3 METHOD

## 3.1 Physical Setup

Our physical system configuration is shown in Figure 1. We employ two stationary lamps with script-controllable hue bulbs coupled with the Shutter robot model. We also use a Linux laptop with GPU capabilities for its ROS compatibility and ability to train our neural network models. Note that this setup is in a controlled environment (university laboratory room), which reduces some of the complexity that would come from developing a system in-the-wild. At the moment, the lights are fixed and changes in lighting are limited to brightness only, but our approach would enable fairly easy incorporation of other lighting decision options such as color.

## 3.2 Problem Definition

We define our formulation of determining the best lighting policy as a reinforcement learning problem. Specifically, the problem can be defined by the following Makrov Decision Process parameters:

- $S = \{I \times FP \times RP \times B\}$ describes the state space of the overall lighting photography system. Here, $I$ represents potential $64 \times 64$ resized cropped images of a face from the robot's RGB-D camera, $FP$ is the set of potential $(x, y, z)$ poses of the face in the robot's base coordinate frame, $RP$ is the set of possible robot poses, and $B$ is the tuple corresponding to potential brightness levels of the two light bulbs $(b_1, b_2)$.
- $A = \{a_1, a_2\}$ describes the action space of the lighting system, where $a_1$ and $a_2$ correspond to the potential commanded light brightness levels for the different bulbs. These brightness levels range from $0-240$, and are discretized by intervals of 30 for the reward network implementation.
- $R$ describes the reward function calculated on the cropped face image from the color camera feed. For our initial model the reward consisted solely of image entropy, which is a well-established metric of image contrast and sharpness computed on the histogram of an individual image. However, for our final model we implemented a more complex reward function that seemed to more accurately capture the aesthetics of subjectively well-lit portrait photographs

## 3.3 Reward Function

We tried a number of different image metrics to serve as a reward for our network:

- Entropy - The entropy of an image is a measure of the degree of randomness in an image, using information (Shannon) entropy. This proved to be the most useful of our image metrics.
- BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator) - Uses Natural Scene Statistics (NSS) to create an image quality assessment without any ground knowledge or supervision. This metric is specialized for recognizing blurring and was not particularly effective for our case.
- DeepIQA - Uses a deep convolutional neural network to get a sense of the quality of an image. This metric, as above, was not particularly effective in our specific domain, where the independent variable is lighting, rather than blurring or some other aspect of the photo.
- Color and Brightness Contrast - These metrics measure the color and brightness contrast of the image. These metrics proved to be useful in our domain, particularly when looking at the local contrast around the face of the person being photographed.

In the end, we used a combination of entropy and brightness and color contrast, with a greater weight towards the entropy.

## 3.4 Data Collection

For our data collection phase, we cycled through a number of different robot poses (camera angle), where for each robot pose, we cycled through 81 different combinations of lighting (each bulb had every value in increments of 30 from 0-240, for a total of 9 X 9 = 81 poses). It is important to note that although each of the poses varied the height and direction the robot was facing, as can be seen in Figure 3, the variation was not drastic since Shutter's view was limited by its surrounding frame. Then, for each of these arrangements, we collected data from two different people. The data itself consisted of all the necessary information for our state space and reward metrics, as discussed above in our problem formulation.

## 3.5 Solution Approach

Due to the vastness of the state space (64*x*64 images, possible robot poses, possible face poses, etc.), we could not run traditional policy learning without some sort of compression of the state space as well as the continuous action space. Consequently, we used a deep learning approach for value approximation, and in the actor-critic implementation, for action selection as well.

*3.5.1 Reward Network Implementation.* We designed a neural network which feeds the cropped face images into a convolutional neural network (CNN), and then feeds the output of the CNN along with the other state space parameters into a feed-forward network to more-generally approximate the instantaneous reward function across the state space. A visual diagram of the entire reward network is shown in Figure 2.

We also discretized the action space such that each light bulb was restricted to brightness levels separated by 30 units in the range [0,240], which led to a total action space set size of 81. We feel that this discretization can be justified from a performance standpoint because lighting changes on the order of about 10 unit changes in brightness were found to be fairly small from qualitative testing. However, adding many more than 81 brightness pairs to the action space would slow down the interaction considerably, as 81 action combinations already prompts a system performance time on the order of about 2-3 seconds. Therefore, from the perspective of our current model, we believe this trade-off is one that's worthwhile, or at the very least worth exploring.

We implemented a CNN and feed-forward neural network through the Keras API. Specifically, the CNN has various layers of 2D pooling, ReLU activation, batch normalization, and dropout regularization to extract a vector of features from $64 \times 64$ RGB images. The feed-forward network is a more shallow network with ReLU activation throughout.

Finally, in order to optimize the lighting for a picture, the system takes the current state information and makes the argmax action (lighting combination) based on the instantaneous reward neural network over the 81 potential actions.

In this implementation, when dealing with multiple faces, the network selects the action with the highest reward for any once face, which may not necessarily lead to the best group photo lighting.

*3.5.2 Actor-Critic Implementation.* A limitation of our reward-approximator network approach, and reward network more generally, is that it is only valid for discrete action spaces. Our approach in particular requires that the program iterate over all possible actions to find the argmax reward value. While the discretization of the action space in our problem is relatively inconsequential, and the argmax iteration not prohibitively costly, we thought it worthwhile to try an approach in which a deep network directly outputs the estimated best action in the continuous action space.

Using the network architecture described above as a critic network, we implemented an actor network trained on our captured dataset, using the critic network gradients with respect to actions for training. The actor network has an identical architecture to the critic, except for an added sigmoid layer in the feed-forward network and a 2-dimensional vector output representing the action output. The output represents the 2-dimensional, continuous, bounded [0, 240] action space of the lighting system. In this implementation, when multiple faces are present in the input image, the network chooses the average of the recommended actions for each face.

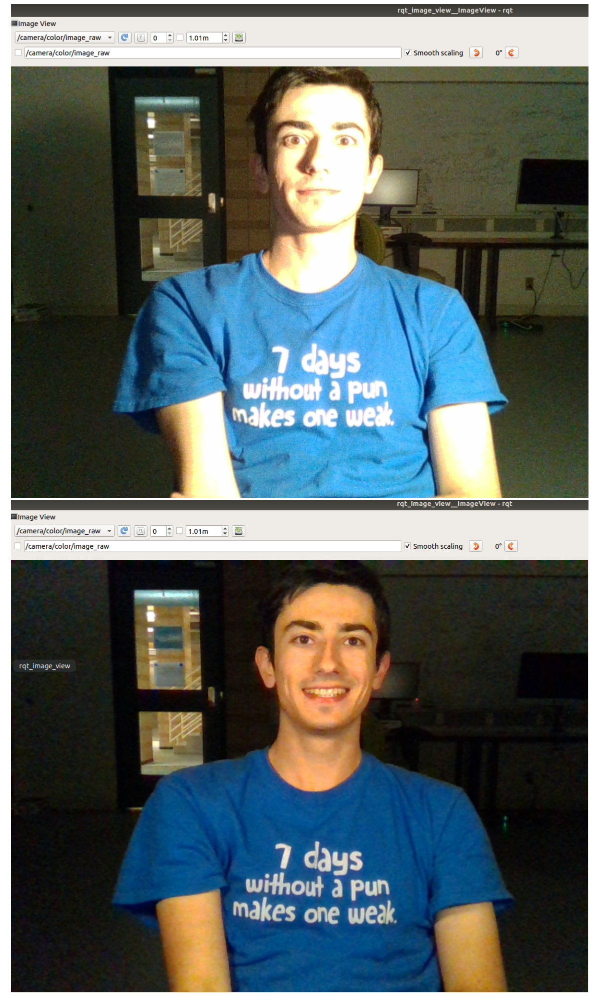


**Figure 4: Lighting conditions before (top) and after our (bottom) network policy calculation and light command calculated from the first version of our instantaneous reward network with limited data**

## 4 EXPERIMENTS

We have successfully created the infrastructure to collect and save images for different robot poses for all the different combinations of lighting for our two light bulbs. As a result, we have collected a preliminary data set measuring states, actions, and rewards (a total of roughly 16,400 samples) captured over a variety of poses, environments, and actions, for batch reinforcement learning. This data was used to train and evaluate our reward network and actor-critic implementations.

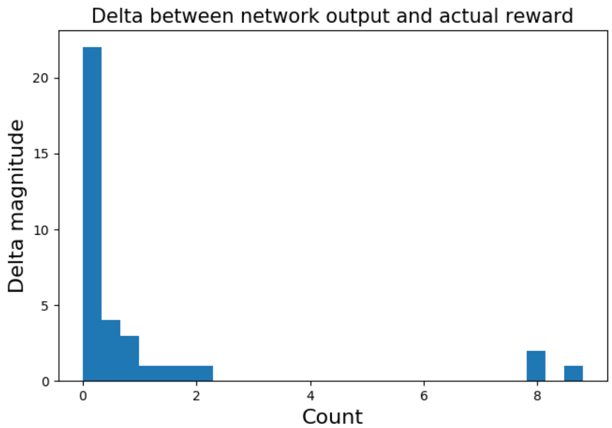


**Figure 5: A histogram summarizing the distribution of differences between our neural network reward approximator and the actual instantaneous reward as calculated by our reward function/metrics**

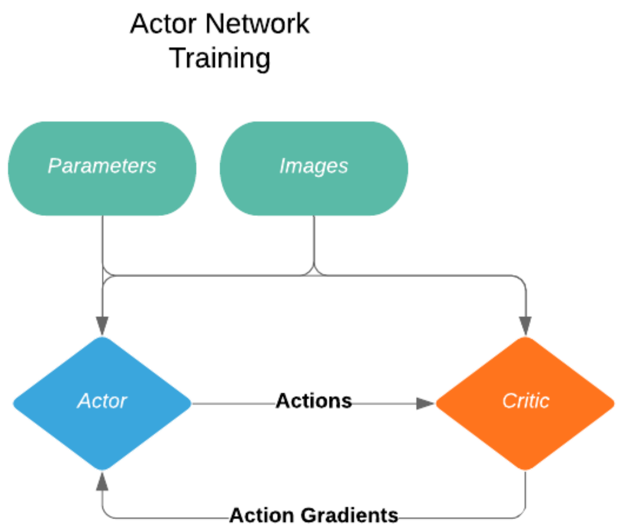


**Figure 6: A flowchart which describes the high-level model surrounding our actor-critic implementation. Note that "parameters" refers to all non-image inputs to the network, including the poses of Shutter and the human, as well as the bulb brightness levels.**

### 4.1 Initial Reward Network Implementation

We first trained a model based on a one pose only dataset (with roughly 500-1000 data points), as a proof of concept for our general network architecture. We then evaluated the reward estimator network qualitatively, and received generally favorable results. An example can be found in Figure 4. It is important to note that this network was limited in terms of poses that Shutter could take pictures from and even the poses at which the human could take, given that the dataset was so limited. This was our first iteration of the reward network implementation.

### 4.2 Final Reward Network Implementation

When then evaluated on a simple train-test split of our more thoroughly collected data over multiple poses, and the reward network model achieved a final mean average percentage error of 5.9% on the reward value of the input pose, image, and lighting data. Furthermore, the network seemed to map the instantaneous reward function quite well quantitatively, as discovered through exploring different combinations of lighting and Shutter pose and calculating the differences in calculated vs. observed reward. The results from our experiment are as shown in Figure 5, with differences between the network output and the actual reward being primarily centered at zero with high skew due to a few very large outliers ($N = 36, mean = 1.823, SD = 5.014$).

In addition to being quantitatively promising, the reward network model was implemented into our ROS architecture for shutter as a reward estimator, with qualitatively favorable results in subjectively well-lit portrait photos. It is important to point out, that our network was not perfect from the multiple outliers in the dataset. There were also test runs qualitatively where extreme lighting choices would be chosen when they were clearly not optimal, leading us to believe there may have been two limitations to our model-namely, our dataset may not have been representative enough of the underlying data, and our network may not have been able to fit our data completely based on our reward metrics. Nevertheless, in general, we feel we have successfully trained our model on a limited collected dataset and tested it by having it optimize the lighting based on a pre-programmed initial state (i.e. with bad lighting).

### 4.3 Actor Critic Implementation

We then implemented an actor-critic network as an alternative to the reward network approximator mentioned previously. The general actor-critic model implementation we used for our purposes is shown in Figure 6. The critic network used in our actor critic implementation was identical to that mentioned above and so had identical performance on the test set. To evaluate the actor network, we used the critic network to label and assess the maximum possible rewards for each input in the test set. The actor network was then used to predict best actions for the test set. Finally, these predicted actions were assessed using the critic network. The reward value of the actor-predicted actions were, on average, within 1.1% of the critic-labelled max rewards.

This result suggests the actor was accurately implementing the best policy described by the critic network on a continuous space without manual iteration over possible actions. The actor has also been successfully implemented into our system as a possible decision maker.

It is important to note, however, that like the reward network implementation, there were some problems in constructing the actor-critic model. In particular, the network qualitatively seems to prefer extreme values for lighting (e.g. [240, 240], with both bulbs being maximally bright), even though in the vast majority of scenarios, maximum lighting is clearly not the best way to achieve a portrait photo in a controlled setting.

## 5 CONCLUSION AND FUTURE WORK

We were able to develop and train networks that could both accurately estimate the reward (based on the image metrics described above) and learn a lighting policy based on that reward learning. We incorporated these networks into a ROS infrastructure and hardware setup that allowed us to achieve our goal of building an interactive and automatic lighting adjustment system for the Shutter robot photographer.

Looking back on our game plan from the proposal, there are certainly changes that we made from the initial plan of action in order to better tackle our proposed problem, especially in terms of our model formulation. Reformulating the problem as one of reinforcement learning prompted us to draw inspiration from existing reinforcement learning techniques like DQN to inform our approach. Though we did still employ a CNN to extract features from the cropped face images from Shutter's camera as part of our system, we broadened the scope of our model to fit the reinforcement learning framework and better suit the constraints of the automated lighting problem.

There are many improvements that could be made to our system to make it more feasible in real-world applications. One such improvement could be in algorithmic design. As discussed previously, reward approximation networks may suffer from extrapolation error when constrained to precollected batches of data as opposed to being allowed to explore states and actions. Fujimoto et al.'s Batch-Constrained Q-Learning (BCQ) algorithms [6] seek to reduce extrapolation error in learning by accounting for the distribution of the batch data relative to the range of possible states. This work suggests that it could be possible to build a batch reinforcement learning system such as ours while still maintaining favorable performance in more unfamiliar environments.

For some applications, it may even be better to take a more traditional exploration-based reinforcement learning approach. For example, if the lighting and photography setup was more mobile and intended to be moved and restructured often, it would make the most sense for the system to explore the new environment and lighting arrangement to learn more quickly about its environment rather than capturing and storing large batches of new data for each environment or setting.

One potential problem for our actor system was that we only worked with a few simple image quality metrics like entropy and local brightness. These metrics qualitatively measured portrait quality well, but may have posed difficulties to the network due to their sometimes unintuitive preferences (for example, an image lit only from one side may have almost the same metric score as one lit evenly from both sides). Incorporating a network trained more specifically on professionally-rated portrait photos (rather than many types of photos) might be better at discriminating the optimal result image. A future version of our system may incorporate or emulate these new techniques for determining the reward value of the captured images.

Further, the ways that we chose to deal with multiple faces present in the input - either taking the best action for any face, or taking the average of the recommended actions for each - were chosen fairly arbitratily. Although they had good results in our limited tests, future work may choose to find other, more robust ways of optimizing lighting for multiple faces at once. One possibility would be to select the action that maximizes the total estimated reward. For designing and training an actor network, one could label group photos with actions and global (rather than local) rewards, and design the actor network so that it could take multiple faces as input and process them as part of the same image.

## REFERENCES

[1] Ismail Taha Ahmed, Chen Soong Der, and Baraa Tareq Hammad. 2017. A Survey of Recent Approaches on No-Reference Image Quality Assessment with Multiscale Geometric Analysis Transforms.
[2] Lexie Carbone. 2018. *The Best Instagram Filters in Photography Apps.* Retrieved November 10, 2019 from https://later.com/blog/instagram-filters/
[3] Sreenithy Chandran and Suren Jayasuriya. 2019. Adaptive Lighting for Data-Driven Non-Line-of-Sight 3D Localization and Object Identification. arXiv:eess.IV/1905.11595
[4] Hui Fang and Meng Zhang. 2017. Creatism: A deep-learning photographer capable of creating professional work. arXiv:cs.CV/1707.03491
[5] Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. 2019. Benchmarking Batch Deep Reinforcement Learning Algorithms. arXiv:cs.LG/1910.01708
[6] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 2052–2062. http://proceedings.mlr.press/v97/fujimoto19a.html
[7] Hillary K. Grigonis. 2018. *What is portrait mode? How tech helps smartphones capture a better you.* Retrieved November 10, 2019 from https://www.digitaltrends.com/photography/what-is-portrait-mode/
[8] J. Kim, A. Nguyen, and S. Lee. 2019. Deep CNN-Based Blind Image Quality Predictor. *IEEE Transactions on Neural Networks and Learning Systems* 30, 1 (Jan 2019), 11–24. https://doi.org/10.1109/TNNLS.2018.2829819
[9] Sean Lewis. 2019. *Eva The Robot Photographer-For-Hire Now Photographs Weddings.* Retrieved November 8, 2019 from https://www.slrlounge.com/eva-the-robot-photographer-for-hire-now-photographs-weddings/
[10] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. arXiv:cs.LG/1509.02971
[11] Kwan-Yee Lin and Guanxiang Wang. 2018. Hallucinated-IQA: No-Reference Image Quality Assessment via Adversarial Learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 732–741.
[12] A. Mittal, A. K. Moorthy, and A. C. Bovik. 2012. No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Transactions on Image Processing* 21, 12 (Dec 2012), 4695–4708. https://doi.org/10.1109/TIP.2012.2214050
[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* (2015), 518–529. https://doi.org/10.1038/nature14236
[14] Maisy Mok. 2019. *This robot photographer will take your profile pic for you.* Retrieved November 4, 2019 from https://www.cnn.com/travel/article/worlds-first-robotic-photographer-intl-hnk-scli/index.html
[15] A. K. Moorthy and A. C. Bovik. 2010. A Two-Step Framework for Constructing Blind Image Quality Indices. *IEEE Signal Processing Letters* 17, 5 (May 2010), 513–516. https://doi.org/10.1109/LSP.2010.2043888
[16] Du-Yih Tsai, Yongbum Lee, and Eri Matsuyama. 2007. Information Entropy Measure for Evaluation of Image Quality. *Journal of Digital Imaging* 21, 3 (2007), 338–347. https://doi.org/10.1007/s10278-007-9044-5
[17] Poom Wettayakorn, Siripong Traivijitkhun, Ponpat Phetchai, and Suppwong Tuarob. 2018. A Deep learning methodology for automatic assessment of portrait image aesthetic quality. In *Proceedings of the 15th. International Joint Conference on Computer Science and Software Engineering (JCSSE '18).* IEEE. https://doi.org/10.1109/JCSSE.2018.8457381