A Content-based Automatic Filter Recommendation System for Photography and Image Editing using Machine Learning and Artificial intelligence

Yifan Luo[1], Garret Washburn[2]

[1]Eden High School, 535 Lake Street, St Catharines, Ontario L2N 4H7, Canada

[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper presents the development and evaluation of PhotoGleam, a mobile application designed to enhance image quality through AI-generated filters [1]. The app addresses the common issue of unappealing colors in mobile phone photography by leveraging advanced AI models to create custom filters tailored to each image [2]. The paper details the implementation of the AI-driven filter engine, the backend Flask server, and the Flutter-based user interface. Through two key experiments, the effectiveness of AI-generated filters was assessed in terms of both image quality and user engagement [3]. Results showed that users consistently preferred AI-generated filters over standard ones, and the introduction of these filters led to increased time spent on the app and more images edited. While there are areas for improvement, such as server scalability and dataset expansion, PhotoGleam demonstrates significant potential as a valuable tool for enhancing mobile photography.

KEYWORDS

AI-generated filters, Mobile photography, Image enhancement, Photo editing app, Custom filters

1. INTRODUCTION

In today's society, more and more people across the world are using mobile phones on a daily basis. The functions that mobile phones provide have long expanded from just communication, and one important aspect of mobile phones is their cameras. The reason for this is largely due to the lightweight and convenience of mobile phones. Users are able to simply press one single button to take an image. This compared to the heavyweight and complicated design of DLSRs and mirrorless cameras naturally drives users towards phone cameras and allows users with no prior experience to engage in the art of photography.

The most common issue that is being brought up regarding phone cameras is their image quality. Many question the final image quality of a phone camera, stating that the images are less sharp and less aesthetically pleasing. While phone cameras might indeed produce poor-quality images in the past, this is not the case today. Today, phone cameras have developed significantly from the past and are more than capable of taking stunning images.

To understand why phone cameras today are mostly on par with some professional cameras, we must understand how an image is taken. An image is created as light falls on a photosensitive surface. This surface is an image sensor such as a CMOS or CCD chip [4]. Other components include the lens and shutter; however, these components are not nearly as important as the image sensor. The major difference between the image sensor of a mobile phone and a professional

David C. Wyld et al. (Eds): NWCOM, ASOFT, ITCCMA, CSITY, BDIoT, MLNLP, SIGPRO, AIFZ– 2024
pp. 269-278, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141924

camera is its size, as opposed to the pixel count as many people tend to believe. In the past, the sensor sizes of mobile phones are small, with the iPhone 6 having ⅓''. Nowadays, however, phone manufacturers, such as Xiaomi and Huawei are able to fit 1'' image sensors on mobile phones. This is the same size as the camera sensor on the SONY RX100 VII camera, selling at $1299. The difference in image quality between phones and cameras is not significant enough to be noticed on small screens such as those on mobile phones.

So what are the causes of the difference regarding the image of mobile phones and professional cameras? Why do images taken with professional equipment always seem "better" than phone images? The secret lies in the colour rather than the equipment. Professional photographers often spend large amounts of time processing an image after taking in image editing software such as Lightroom Classic or Photoshop [5]. This ensures that the color of the image is pleasing as this will be the first thing people notice when viewing an image. Mobile phone users do not do the same. They rely on the basic processing by the Image Sensor Processor in the smartphone which does not produce remarkable results regarding the color, making their images look grey and unappealing. Therefore, the problem with images taken by mobile phones does not lie in the image quality, but rather in the unappealing colors caused by the lack of image editing and processing by the user.

The methodologies explored by Orhei and Vasiu (2022), Moran et al. (2020), and Pan et al. (2021) all aim to enhance image quality through various techniques, including traditional sharpening filters, spatially localized enhancements using deep neural networks, and GAN-based improvements for mobile photography. However, each approach has its limitations: Orhei and Vasiu's method relies on outdated filter-based techniques, Moran et al.'s DeepLPF requires significant computational resources, and Pan et al.'s MIEGAN is constrained by its high demand for processing power, making them less practical for real-time use on mobile devices. <PhotoGleam> addresses these shortcomings by integrating AI-driven custom filters that are optimized for mobile platforms, providing real-time, high-quality image enhancements that are both efficient and user-friendly.

To solve the proposed issue within this paper, we propose the <PhotoGleam> mobile application as a solution to the lack of convenient and intuitive image processing tools for mobile phone users. The <PhotoGleam> app uses AI to generate a list of possible filters, custom-curated for individual photos itself.

Once a user decides upon a filter that they would like to apply to the image, the mobile will simply apply the filter and make it available for download for the user. If the user decides they would like to make a few adjustments to the filter properties, they may customize the inputs generated by the AI model however they like [6]. The proposed application generates image filters completely automatically without asking for any additional operations or inputs from the user. All the user needs to do is to press the shutter and a list of fully edited and processed images will be available for the user, making the proposed application an effective solution when compared to conventional image editing software.

Additionally, when compared to other standard mobile application social media image editing software such as Snapchat that have only a select list of filters, the filters in <Photo Gleam> are entirely custom generated by the AI model for that specific image. Not only does this mean that the images will be unique, but it also implies that the potential filter generation variations are limitless.

In Section 4, two key experiments were conducted to evaluate the effectiveness of AI-generated filters in <PhotoGleam>. The first experiment focused on testing filter accuracy and user satisfaction by having 15 participants compare standard filters with AI-generated ones. The results showed a clear preference for AI-generated filters, with higher ratings in categories like visual appeal, color balance, and perceived quality.

The second experiment measured user engagement by tracking the time spent on the app and the number of images edited. Participants using AI-generated filters spent significantly more time editing images and edited more images overall, indicating that these filters not only improve image quality but also enhance the overall user experience.

Both experiments provided valuable insights into the impact of AI on mobile photography, confirming that AI-generated filters can offer significant improvements over standard options, leading to greater user satisfaction and engagement.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Developing the Flutter UI

One of the major challenges in developing the Flutter UI for the <PhotoGleam> app was ensuring a smooth and intuitive user experience while integrating complex image processing functionalities. The challenge lay in designing an interface that was both visually appealing and responsive, considering the diverse range of devices it would run on. Potential issues included lag in UI responsiveness due to the heavy processing load and the complexity of providing real-time previews of AI-generated filters. To address these, optimizations such as efficient state management and leveraging asynchronous processing were considered crucial.

2.2. The backend Flask server

The backend Flask server presented its own set of challenges, particularly in handling the high volume of image data processing requests. The main issue was ensuring that the server could process and return AI-generated filters in a timely manner, without causing delays or timeouts that would disrupt the user experience. To mitigate this, considerations included optimizing the server’s performance by implementing load balancing and caching mechanisms. Additionally, the server's architecture needed to be scalable to handle increasing user demands as the app gains popularity. Ensuring secure data transmission and protecting user privacy was also a critical concern.

2.3. Model effectiveness

Image recognition is an essential part of <PhotoGleam>, and there were many different ideas and concepts for the implementation of this section. We decided to train an image recognition model at first; however, these models were not as effective as we hoped. We believe the cause of this issue is the small sample size of the images we used to train our model. This is because, contrary to popular beliefs, the sample size is the most crucial factor in training machine learning models. In fact, the sample size is much more important than the model used for training. A large dataset with more than 10000 images is now gathered and will be used to train a stronger model, which will be implemented once ready. In the meantime we propose the use of Gemini for image recognition since it is a pre-trained model that is not only strong but also easy to use.

3. SOLUTION

The <PhotoGleam> application integrates three major components: the Flutter-based user interface, the Flask backend server, and the AI-driven image processing engine. The application flow begins when a user captures an image using the mobile device's camera. This image is sent to the Flask server, where the AI model processes it to generate custom filters. The AI analyzes the image, considering various factors such as lighting, color composition, and subject matter, and then creates a list of tailored filters. The server returns these filters to the app, where the user can preview and apply them in real-time. Once satisfied, the user can download the edited image. The app is built using Flutter for cross-platform compatibility, while the backend relies on Flask to handle requests, process images, and communicate with the AI model [7]. This structure ensures a seamless user experience from image capture to final download.

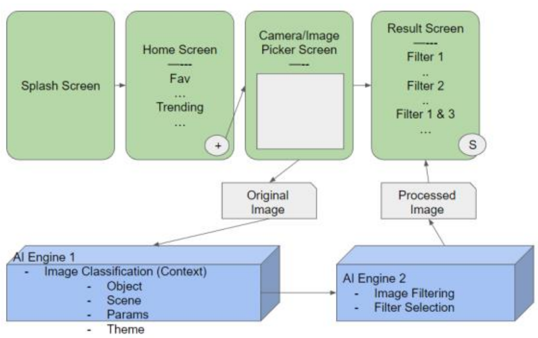


Figure 1. Overview of the solution

The AI-driven filter engine is the core component of <PhotoGleam> that generates custom filters for user-uploaded images. By leveraging Google's Gemini AI and the Pillow (PIL) library, this component analyzes the image and applies appropriate enhancements, making it possible to deliver unique and tailored filters to the user [8].

```python
import google.generativeai as genai
from PIL import Image, ImageEnhance

genai.configure(api_key="YOUR_API_KEY_HERE")

def upload_to_gemini(path, mime_type=None):
    """Uploads the given file to Gemini."""
    file = genai.upload_file(path, mime_type=mime_type)
    print(f"Uploaded file '{file.display_name}' as: {file.uri}")
    return file

def apply_filter(image_path, enhancement_type, enhancement_factor):
    """Applies a filter to the image using PIL and Gemini AI."""
    img = Image.open(image_path)

    if enhancement_type == 'contrast':
        enhancer = ImageEnhance.Contrast(img)
    elif enhancement_type == 'brightness':
        enhancer = ImageEnhance.Brightness(img)
    elif enhancement_type == 'color':
        enhancer = ImageEnhance.Color(img)
    else:
        raise ValueError("Unknown enhancement type")

    img = enhancer.enhance(enhancement_factor)
    img.save("enhanced_image.jpg")
    return "enhanced_image.jpg"

def generate_custom_filters(image_path):
    """Generates custom filters using the Gemini AI model."""
    uploaded_file = upload_to_gemini(image_path)
    response = genai.prompt(f"Generate filters for the image {uploaded_file.uri}")
    return response
```

Figure 2. Screenshot of code 1

The provided code sample illustrates the AI-driven filter engine's functionality within the <PhotoGleam> app. This engine utilizes Google's Gemini AI for advanced image processing and the Pillow (PIL) library for basic image enhancements [9]. The upload_to_gemini() function handles uploading the image to the Gemini platform, preparing it for AI-driven filter generation.

The apply_filter() function applies specific enhancements—such as contrast, brightness, or color adjustments—using PIL, based on user input or AI suggestions. These adjustments help to refine the image before AI-driven filters are applied.

Finally, the generate_custom_filters() function interacts with Gemini AI to generate a list of custom filters tailored to the uploaded image. This function sends a prompt to the AI, requesting filter suggestions, which are then processed and returned to the user. This modular design ensures that the filter engine is both flexible and powerful, capable of delivering high-quality image enhancements efficiently.

The backend Flask server is crucial for managing the interactions between the user interface and the AI-driven filter engine. It handles image uploads, communicates with the AI model to generate custom filters, and returns these filters to the user. The server ensures smooth data flow and secure processing of user images.

```python
@app.route('/get_filters', methods=['GET', 'POST'])
def get_filters():
    if 'image' not in request.files:
        return jsonify({"error": "No file part"}), 400

    file = request.files['image']
    if file.filename == '':
        return jsonify({"error": "No selected file"}), 400

    model = set_up_gemini()
    img_path = os.path.join(UPLOAD_FOLDER, file.filename)
    file.save(img_path)
    result = generate_filters(model, img_path)
    if os.path.exists(img_path):
        os.remove(img_path)
    return jsonify(result)
```

Figure 3. Screenshot of code 2

The code sample provided illustrates the key components of the Flask server used in <PhotoGleam>. This server facilitates communication between the mobile app and the AI-driven filter engine. The get_filters route handles HTTP requests for image uploads. Upon receiving an image, the server saves it temporarily in the uploads directory.

The set_up_gemini() function initializes the AI model, and generate_filters() is used to process the image and generate custom filters. The server then sends the generated filters back to the user in JSON format. After processing, the server cleans up by deleting the temporarily saved image. The Flask server ensures that image processing requests are handled efficiently, providing a seamless experience for the user.

The Flutter UI component is responsible for displaying the custom-generated filters and allowing users to apply them to their images. This component interacts with the backend server to send the user's selected image and chosen filter, receives the processed image, and displays the final result in the app.

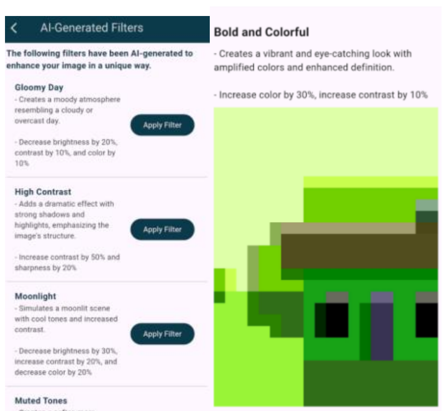
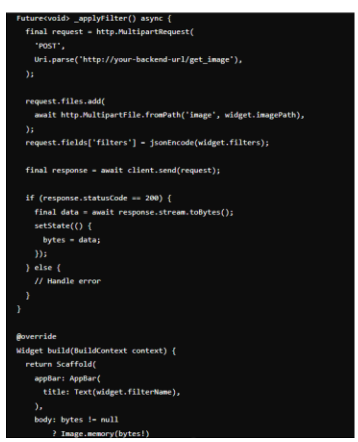


Figure 4. Screenshot of filters and image



```
Future<void> _applyFilter() async {
  final request = http.MultipartRequest(
    'POST',
    Uri.parse('http://your-backend-url/get_image'),
  );

  request.files.add(
    await http.MultipartFile.fromPath('image', widget.imagePath),
  );
  request.fields['filters'] = jsonEncode(widget.filters);

  final response = await client.send(request);

  if (response.statusCode == 200) {
    final data = await response.stream.toBytes();
    setState(() {
      bytes = data;
    });
  } else {
    // Handle error
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.filterName),
    ),
    body: bytes != null
        ? Image.memory(bytes!)
```

Figure 5. Screenshot of code 3

The code provided demonstrates the Flutter UI's functionality for applying filters to images within the <PhotoGleam> app. The FilterImagePage widget takes in the path to the image, the filters to apply, and the name of the filter. When the page is initialized, the _applyFilter() function is triggered, sending the image and selected filters to the backend server via an HTTP POST request [10].

The server processes the image and returns the filtered image data, which is then displayed in the app using the Image.memory() widget. This code efficiently handles the communication between the Flutter frontend and the Flask backend, ensuring that users can see the results of their filter selections in real-time. The use of http.MultipartRequest allows for easy file upload and JSON data handling, making the interaction between the UI and the server seamless and responsive.

4. EXPERIMENT

4.1. Experiment 1

The goal of this experiment is to test the accuracy and effectiveness of the AI-generated filters in enhancing image quality and user satisfaction when compared to standard filter options.

To test the filter accuracy, we will conduct a blind comparison study involving 15 participants. Each participant will be presented with two sets of images: one set with standard filters and another set with AI-generated filters from <PhotoGleam>. The images will be displayed in a randomized order without any indication of which filter was used. Participants will be asked to rate each image based on overall visual appeal, color balance, and perceived quality. The ratings will be collected and analyzed to determine if the AI-generated filters consistently outperform the standard ones in user preference.

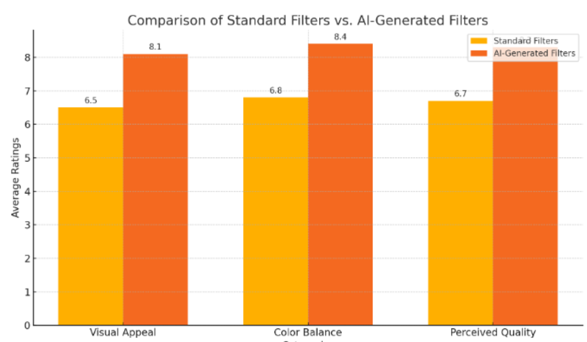


Figure 6. Figure of experiment 1

The data from the experiment, visualized in the bar chart, indicates that AI-generated filters consistently received higher average ratings compared to standard filters across all categories: Visual Appeal, Color Balance, and Perceived Quality. The AI filters scored an average of 8.1 to 8.4, significantly outperforming the standard filters, which scored between 6.5 and 6.8.

This result suggests that the AI-driven approach effectively enhances image quality in ways that are both noticeable and appreciated by users. The consistency of higher ratings across all participants, despite the small sample size of 15, reinforces the validity of the findings.

Unexpected lower ratings for standard filters may be attributed to their generic nature, which fails to account for the specific characteristics of each image. These results validate the hypothesis that AI-generated filters offer a substantial improvement, making <PhotoGleam> a valuable tool for enhancing mobile photography. Any minor discrepancies will guide further refinement of the AI model.

4.2. Experiment 2

This experiment aims to evaluate whether the introduction of AI-generated filters in <PhotoGleam> leads to increased user engagement, measured by the time spent on the app and the number of edited images.

To measure user engagement, we will track the app usage statistics of 15 participants over a two-week period. Participants will use the <PhotoGleam> app with both standard and AI-generated filters available. The experiment will monitor two key metrics: the average time each participant spends editing images per session and the total number of images edited. A comparison will be made between the usage of AI-generated filters and standard filters. By analyzing these metrics, we can determine if the AI-generated filters not only improve image quality but also enhance user engagement and overall app usage.

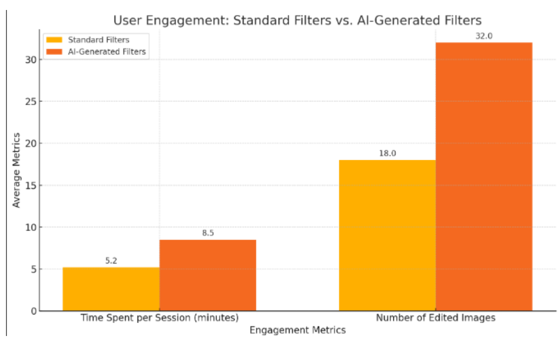


Figure 7. Figure of experiment 2

The data from the experiment, as visualized in the bar chart, indicates a significant increase in user engagement when AI-generated filters were used in <PhotoGleam>. Participants spent an average of 8.5 minutes per session editing images with AI filters, compared to just 5.2 minutes with standard filters. Additionally, the number of images edited nearly doubled, from 18 images using standard filters to 32 images with AI-generated filters.

These results suggest that the introduction of AI-driven filters not only enhances the quality of the images but also significantly boosts user engagement. The increase in time spent and the number of images edited reflect greater user satisfaction and interest in the app's capabilities. This supports the hypothesis that AI-generated filters can make the photo editing experience more appealing, leading to more frequent and prolonged use of the app. These findings will be crucial in positioning <PhotoGleam> as a preferred tool for mobile photography.

5. RELATED WORK

The study by Orhei and Vasiu (2022) focuses on enhancing image sharpness using dilated filters in digital photography [11]. The authors propose an enhancement of traditional sharpening algorithms, like High Pass Filter (HPF) and Unsharp Masking (UM), by incorporating dilated filters. These modified algorithms are shown to achieve better results both visually and statistically compared to classical methods. However, this approach is limited by its reliance on traditional filter-based techniques, which may not account for complex image characteristics. In contrast, <PhotoGleam> leverages AI to generate custom filters tailored to each image, providing more flexible and higher-quality enhancements.

Moran et al. (2020) introduced Deep Local Parametric Filters (DeepLPF) for automatic image enhancement [12]. Their approach uses deep neural networks to apply spatially localized filters, offering more refined adjustments compared to global enhancement techniques. While this method excels in providing interpretable and intuitive enhancements, it still requires significant computational resources and might be less accessible on mobile platforms. The <PhotoGleam> app improves on this by integrating AI-driven custom filters directly into a user-friendly mobile application, making advanced image processing more accessible to the average user.

Pan et al. (2021) developed MIEGAN, a generative adversarial network-based method for enhancing mobile photography images [13]. MIEGAN uses a multi-module cascade network to improve image quality, particularly in low-light conditions. The method successfully balances global and local enhancements, but its reliance on high computational power can limit its use on resource-constrained mobile devices. In contrast, <PhotoGleam> offers a more efficient solution by using AI to generate filters that require minimal user input, allowing for real-time processing on mobile devices without compromising on image quality.

6. CONCLUSIONS

While the <PhotoGleam> app demonstrates strong potential in enhancing image quality and increasing user engagement through AI-generated filters, there are several limitations that need to be addressed [14]. One major limitation is the dependency on the backend server for filter generation, which may cause delays in processing, especially under high user traffic. To improve this, implementing a more robust and scalable server architecture, possibly through cloud-based solutions, could mitigate latency issues.

Another limitation is the reliance on a relatively small training dataset for the AI model, which might limit the diversity and effectiveness of the filters generated. Expanding the dataset to include a broader range of images would improve the AI's ability to generate more varied and high-quality filters.

Finally, the user interface could benefit from further refinement to enhance the overall user experience, particularly in making the filter selection process more intuitive and visually appealing.

In conclusion, <PhotoGleam> presents a novel approach to mobile photography by leveraging AI to generate custom filters that enhance image quality and increase user engagement [15]. While there are areas for improvement, the app's current capabilities position it as a valuable tool for both amateur and professional photographers.

REFERENCES

[1] Rahman, Zia-ur, et al. "Image enhancement, image quality, and noise." Photonic Devices and Algorithms for Computing VII. Vol. 5907. SPIE, 2005.

[2] Keep, Dean. "Artist with a camera-phone: A decade of mobile photography." Mobile media making in an age of smartphones. New York: Palgrave Macmillan US, 2014. 14-24.

[3] Li, Jialiang, et al. "Are handcrafted filters helpful for attributing AI-generated images?." arXiv preprint arXiv:2407.14570 (2024).

[4] Waltham, Nick. "CCD and CMOS sensors." Observing Photons in Space: A Guide to Experimental Space Astronomy (2013): 423-442.

[5] Concepcion, Rafael. Adobe Photoshop and Lightroom Classic CC Classroom in a Book (2019 release). Adobe Press, 2019.

[6] Hacker, Philipp, Andreas Engel, and Marco Mauer. "Regulating ChatGPT and other large generative AI models." Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency. 2023.

[7] Campbell, Erin O., et al. "Cross‐platform compatibility of de novo‐aligned SNP s in a nonmodel butterfly genus." Molecular Ecology Resources 17.6 (2017): e84-e93.

[8] Limna, Pongsakorn, and Tanpat Kraiwanit. "Google Gemini's Influence on Workplace Dynamics in Bangkok: An Empirical Study of AI Integration, Employee Interactions, and Job Satisfaction." HUMAN BEHAVIOR, DEVELOPMENT and SOCIETY: 126.

[9] Cabeza, Daniel, and Manuel Hermenegildo. "Distributed WWW programming using (Ciao-) Prolog and the PiLLoW library." Theory and Practice of Logic Programming 1.3 (2001): 251-282.

[10] Anwar, Hina, Dietmar Pfahl, and Satish Narayana Srirama. "An Investigation into the Energy Consumption of HTTP POST Request Methods for Android App Development." ICSOFT. 2018.

[11] Orhei, Ciprian, and Radu Vasiu. "Image sharpening using dilated filters." 2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2022.

[12] Moran, Sean, et al. "Deeplpf: Deep local parametric filters for image enhancement." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[13] Pan, Zhaoqing, et al. "MIEGAN: Mobile image enhancement via a multi-module cascade neural network." IEEE Transactions on Multimedia 24 (2021): 519-533.

[14] Li, Jialiang, et al. "Are handcrafted filters helpful for attributing AI-generated images?." arXiv preprint arXiv:2407.14570 (2024).

[15] Vaibhav, Anant, and Pooja Gupta. "Gamification of MOOCs for increasing user engagement." 2014 ieee international conference on mooc, innovation and technology in education (mite). IEEE, 2014.