|  |
| --- |
| **ABSTRACT** |
| Image style transfer is a deep learning technique for blending the content of one image with the style of another. The output is a new image with the same content as the original content image, but with the style of the style image. This can be achieved by using neural networks to separate and recombine the content and style features of the images.  Image style transfer is an important research content related to image processing in computer vision. Compared with traditional artificial computing methods, deep learning-based convolutional neural networks have powerful advantages. This new method has high computational efficiency and a good style transfer effect. Pre-trained VGG-19 (Visual Geometry Group) neural network models are used to achieve image style transfer.  The main applications of this project are:   1. **Artistic expression**: creating unique and visually appealing images by combining the content of one image with the style of a painting or a famous artist's work. 2. **Graphic design:** using style transfer to generate various design variations and logos. 3. **Photography:** editing photos to give them a desired aesthetic or mood. 4. **Film and video production:** transferring style from reference images to create a consistent visual style for a film or video project. 5. **Virtual Reality and Augmented Reality:** adding artistic styles to VR and AR experiences to enhance their visual appeal. 6. **Printing and publishing:** using style transfer to automatically generate visually appealing layouts and designs for books, magazines, and other printed materials.   The main features of this project are:   1. Use of VGG-19 as a pre-trained convolutional neural network 2. Transfer of style from a reference image to a content image 3. Use of a loss function to optimize image generation 4. Combination of content and style losses for overall transfer 5. Generation of a unique, stylized image that preserves content features. |
| **REFERENCES** |
| **Image Style Transfer Based on VGG Neural Network Model**  **Published in:**2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)  **Conducted by:**IEEE  **Date of Conference:**20-21 August 2022  **Conference Location:**Dalian, China |

**CHAPTER 1**

**INTRODUCTION**

|  |
| --- |
| **1.1 GENERAL INTRODUCTION** |
| Image style transfer is a technique that enables you to blend the content of one image with the style of another, resulting in a new image with the visual traits of both. The VGG-19 convolutional neural network (CNN), which was pre-trained on a sizable dataset of images for image identification tasks, is one well-liked technique for carrying out visual style transfer.  The fundamental idea underlying style transfer with VGG-19 is to use the network's feature maps to gather information about the content and style of the input images. Although the lower-level feature maps that capture the texture and color information of the style picture provide the style information, the higher-level feature maps that capture the semantic information of the input image provide the content information.  The neural style transfer method, which involves maximizing an objective function that balances the content and style of the input images, can be used to execute style transfer using VGG-19. This entails producing a new image that minimizes the separation between the input image's content features and the generated image as well as the style image's features and the generated image.  Create interactive web apps for data science and machine learning tasks using the Python package Streamlit. A user-friendly tool for doing image style transfer on their photographs can be made by combining a Streamlit GUI with the VGG-19 style transfer algorithm. Features like choosing the content and style images, modifying the style transfer parameters, and instantly previewing the resulting image are examples of this. When used together, VGG-19 and Streamlit can offer an easy-to-use method of transferring image style. |
| **1.2 GOAL OF THE PROJECT** |
| This project's objective is to provide a user-friendly solution for image style transfer that makes use of the VGG-19 convolutional neural network and a Streamlit Interface. The application is made to enable users to blend the appearance of one image with the content of another, offering a flexible and user-friendly method for modifying images, expressing one's creativity, and creating designs. Users can use this tool to produce one-of-a-kind, customized photographs that reflect their aesthetic tastes or company identity as well as to improve the visual attractiveness of their already existing images. The goal of this project is to offer a workable and approachable solution for a range of image-related problems by combining the strength of VGG-19 and the ease of Streamlit. |

**CHAPTER 2**

**LITERATURE SURVEY**

|  |
| --- |
| **2.1 STUDY OF SIMILAR WORK** |
| Image style transfer is a method that enables the blending of one image's visual aesthetic with its content. Due to the vast range of uses for this approach, including artistic expression, design, and image manipulation, its popularity has grown. For image style transfer, several strategies have been put forth, each with advantages and disadvantages. The VGG-19, DCGAN, and CAST frameworks are three different visual style transfer algorithms that we evaluate and examine in this study. Due to its capacity to record both content and style information, the CNN-based VGG-19 technique has been frequently employed for image style transfer.  A generative model called DCGAN can create images that are stylistically similar to a reference image while maintaining the integrity of the input image. The CAST framework integrates object awareness with both content and style transfer. It is a framework for content-aware style transfer. By contrasting and evaluating different approaches, we hope to shed light on their benefits and drawbacks as well as suggest possible lines of inquiry for further study on picture style transmission. |
| **2.1.1 EXISTING SYSTEM** |
| The Existing System now uses a variety of methods and strategies for image style transfer. Using pre-trained deep neural networks, like VGG-19, to extract content and style information from the input photos is a typical strategy. These networks serve as feature extractors in the image style transfer process and are trained on big datasets of images for object recognition tasks.  Another strategy is to create new photos with the desired style while keeping the original image's content using generative models like GANs (Generative Adversarial Networks) and VAEs (Variational Autoencoders). These models can either learn to create new styles based on a given reference image or can be trained on datasets of photos with a particular style.  Another strategy is to employ a deep learning framework. The CAST Framework is a multi-stage deep learning strategy for using contrastive learning to transfer the style of one image to another. It can maintain fine features in the image and manage a variety of styles and domains. It improves the capacity to manage several different styles at once and preserve semantic coherence in the transmitted image. |
| **2.1.2 DRAWBACK OF EXISTING SYSTEM** |
| * **Limited diversity in the results of the style transfer:** One of the primary shortcomings of the VGG-19 and DCGAN-based approaches is that they can only produce a limited diversity in the results of the style transfer. This can make the procedure less engaging and creative because the copied photos may end up looking extremely similar to one another. * **Costly to compute**: This is applicable for all algorithms, can be costly to compute, especially when working with high-resolution images or intricate designs. The process of transferring styles may become time- and resource-consuming as a result. * **Absence of fine details:** This is also applicable for all algorithms, occasionally provide transferred images that are hazy or lack fine features, which might lower the output's overall quality. * **Limited control over the style transfer process:** This is present in both VGG-19 and DCGAN-based techniques, which makes it possible that the user won't get the precise outcome they were hoping for. * **Absence of user-friendly GUI:** Some current systems might not have one, which can make it challenging for users to interact with the system and produce the needed outcomes. * **Prototype models only:** Since some of the current systems are only prototypes, they might not be completely optimised or scaled for usage in production. This may restrict their scalability and range of use. |

**CHAPTER 3**

**OVERALL DESCRIPTION**

|  |
| --- |
| **3.1 PROPOSED SYSTEM** |
| The proposed system aims to leverage the advantages of VGG-19 over VGG16 neural network models for image style transfer. The research paper entitled “**Image Style Transfer Based on VGG Neural Network Model”** published at the IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) in 2022 compared the two models and concluded that VGG16 has a simpler architecture, lower computational complexity, comparable performance, and faster training. However, VGG-19 offers a deeper architecture, higher accuracy, increased model capacity, and flexibility for transfer learning, making it a more future-proof choice for the proposed application. Additionally, the proposed system will include a user-friendly GUI and user support materials, making it commercially viable and accessible to a wide range of users. |
| **3.2 FEATURES OF PROPOSED SYSTEM** |
| The proposed system for image style transfer using VGG-19 neural network model will have several key features, including Image Style Transfer, VGG-19 Neural Network Model, User-friendly GUI, Real-time Visualization, Customizable Parameters, Comprehensive Documentation and User Support, and Commercialization Potential. The proposed system will utilize the VGG-19 neural network model, which has a deeper architecture, higher accuracy, increased model capacity, and flexibility for transfer learning. It will also have a user-friendly GUI, customizable parameters, comprehensive documentation and user support materials, and commercialization potential. |
| **3.3 FUNCTIONS OF PROPOSED SYSTEM** |
| The proposed system for image style transfer using VGG-19 neural network model will perform the following functions:   1. **Image Input:** The system will allow users to input a source image and a style image. These images will serve as the input for the style transfer process. 2. **Style Transfer:** The system will apply the style of the style image onto the source image using the VGG-19 neural network model. The style transfer process will involve extracting style features from the style image and transferring them onto the source image, while preserving the content of the source image. 3. **Parameter Customization:** The system will provide users with the ability to customize various parameters, such as the scale of the style transfer, to achieve the desired level of style blending. Users will be able to adjust these parameters in real-time and see the effect on the output image. 4. **Output Image Generation:** The system will generate the style-transferred output image based on the source image, style image, and the adjusted parameters. The output image will reflect the desired style blending between the source and style images. 5. **Multiple Device Compatibility:** The system should be compatible with different devices, including desktop computers, laptops, tablets, and smartphones, allowing users to access and use the application across various platforms. 6. **User-friendly GUI:** The system will have a graphical user interface (GUI) that is intuitive and easy to use. Users will be able to interact with the system through the GUI, input images, adjust parameters, and visualize the output image in real-time. 7. **Download Button for PNG Image:** The system will provide users with a "Download" button in the output interface, which will allow them to save the style-transferred output image in PNG file format. This will provide users with an easy and convenient way to save the generated image for further use. 8. **Documentation and User Support:** The system will be accompanied by comprehensive documentation and user support materials, including user manuals, tutorials, and troubleshooting guides. This will assist users in effectively utilizing the system, understanding its functions, and resolving any issues they may encounter. 9. **Commercialization Potential:** The system will aim to have commercialization potential, with a user-friendly GUI, customizable parameters, and comprehensive user support materials. This will make the system attractive for various applications and industries, providing potential for commercialization and adoption in real-world scenarios. |
| **3.4 REQUIREMENTS SPECIFICATION** |
| 1. **High Accuracy:**   The system should accurately perform image style transfer using VGG-19 model, producing visually appealing and faithful style-transferred output images.   1. **Good Speed:**   The system should process style transfer in a timely manner, without significant delays or lags.   1. **User-friendly Interface:**   The system should have an intuitive and visually appealing GUI that allows users to easily input images, adjust parameters, visualize results, and download output images.   1. **Customizable Parameters:**   The system should provide users with flexibility to customize parameters for style blending.   1. **Multiple Device Compatibility:**   The system should be compatible with various devices and operating systems, such as Windows, macOS, iOS, and Android, to allow users to access and use the application across different platforms.   1. **Robustness and Reliability:**   The system should be able to handle different image types, sizes, and styles without crashing or producing erroneous results.   1. **Documentation and User Support:**   The system should be accompanied by comprehensive documentation and user support materials, including manuals, tutorials, and troubleshooting guides.   1. **Commercialization Potential:**   The system should have potential for commercialization, with scalability, user appeal, and value in various applications and industries. |
| **3.5 FEASIBILITY STUDY** |
| Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system. Feasibility study is one of stage among important four stages of Software Project Management Process. As name suggests feasibility study is the feasibility analysis or it is a measure of the software product in terms of how much beneficial product development will be for the organization in a practical point of view. Feasibility study is carried out based on many purposes to analyse whether software product will be right in terms of development, implantation, contribution of project to the organization etc.  In our proposed system the product is feasibility can be achieved in all four aspects Technical Operational, Economical and Behavioural. |
| **3.5.1 TECHNICAL FEASIBILITY** |
| In Technical Feasibility current resources both hardware software along with required technology are analysed/assessed to develop project. This technical feasibility study gives report whether there exists correct required resources and technologies which will be used for project development. Along with this, feasibility study also analyses technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc. In this proposed system technical feasibility is achieved according to above criteria. |
| **3.5.2 OPERATIONAL FEASIBILITY** |
| In Operational Feasibility degree of providing service to requirements is analysed along with how much easy product will be to operate and maintenance after deployment. Along with these other operational scopes are determining usability of product, determining suggested solution by software development team is acceptable or not etc. The Operational feasibility can be ensured by the proposed system. |
| **3.5.3 ECONOMICAL FEASIBILITY** |
| Economic feasibility the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better results along with increased efficiency. Cost benefit analysis is usually performed for the expected from the proposed system. Since the organization is well equipped with the required hardware, the project was found to be economically feasible and the users who possess a device supports Windows operating system can easily use it. |
| **3.5.4 BEHAVIORAL FEASIBILITY** |
| The proposed system satisfies behavioral feasibility because the system is providing with good and minimalistic GUI which can easily be understand for any end users and its encapsulates the conversion procedure from the users. Hence, it’s easier to operate the system with ease. |
|  |
| *Figure 1: StyleGen Application Logo* |

**CHAPTER 4**

**OPERATING ENVIRONMENT**

|  |
| --- |
| **4.1 HARDWARE REQUIREMENTS** |
| **1. Processor:** Intel i3 7th Gen 3.0 GHz or higher; AMD Ryzen 3 3200G or higher  **2. Hard disk:** 500 GB  **3. RAM:** 6GB or higher  **4. Monitor:** 17” Color Monitor or higher  **5. Mouse:** Microsoft  **6. Keyboard:** Microsoft multimedia keyboard |
| **4.2 SOFTWARE REQUIREMENTS** |
| **1. Operating System:** Windows 10 or higher; Mac OS X 10.11 or higher  **2. Framework:** Flask Framework  **3. Environment:** PyCharm Community Edition 2022.2  **4. Language:** Python 3.8.10, Streamlit 1.18.1, Numpy 1.20.3 and Tensor Flow 2.3.0  **5. Documentation:** Microsoft Word 2010 or higher |
| **4.3 TOOLS AND PLATFORMS** |
| **4.3.1 PYCHARM** |
| PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. |
| **4.3.2 PYTHON 3.8** |
| Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. |
| **4.3.3 STREAMLIT** |
| Streamlit is an open-source Python library that allows data scientists and developers to create interactive web applications for data visualization and machine learning (ML) purposes. It provides a simple and intuitive way to build web applications directly from Python scripts, without requiring extensive web development experience.  With Streamlit, you can easily create interactive user interfaces (UIs) for data-driven applications by writing Python code. Streamlit automatically handles the rendering of the UI components, such as buttons, sliders, and charts, making it easy to create interactive and responsive web applications with just a few lines of code.  Under the hood, Streamlit uses the Flask web framework, which is a lightweight and flexible web framework for Python. Flask provides the underlying web server functionality for Streamlit to handle incoming requests and serve the web application to users. However, as a Streamlit user, you do not need to directly interact with Flask or have knowledge of Flask's intricacies, as Streamlit abstracts the complexity of web development and provides a high-level interface for building web applications using familiar Python syntax.  Streamlit also provides seamless integration with popular Python libraries, such as Pandas, Matplotlib, and TensorFlow, allowing you to easily incorporate data analysis, visualization, and ML functionalities into your web applications.  Streamlit is commonly used in data science and ML workflows for creating prototypes, building dashboards, and sharing data-driven applications with others. It is widely used in industries such as finance, healthcare, and retail, as well as in research and academia, for creating interactive and user-friendly web applications for data analysis and visualization. |
| **4.3.4 FLASK** |
| Flask is a lightweight and flexible web framework for Python that is commonly used for building web applications. It is designed to be simple and easy to use, making it a popular choice for small to medium-sized web projects and prototypes. Flask provides a minimalistic approach to web development, giving developers the flexibility to choose their own tools and libraries to build web applications based on their specific needs.  Flask provides essential features for web development, including routing, request handling, and template rendering, while leaving the decision of how to structure the application and which additional libraries to use up to the developer. This makes Flask highly customizable and allows developers to create web applications tailored to their requirements. |
| **4.3.5 DEEP LEARNING** |
| Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.  In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers. |
| **4.3.6 CNN** |
| In deep learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural  network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.    *Figure 2: CNN*  Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical  functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.  The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc. |
| **4.3.7 VGG-19** |
| VGG-19, also known as **Visual Geometry Group-19**, is a deep convolutional neural network (CNN) architecture for image recognition and computer vision tasks. It was developed by the Visual Geometry Group at the University of Oxford, and it is a variant of the original VGG network, which was introduced by Simonyan and Zisserman in 2014.  VGG-19 consists of 19 weight layers, including 16 convolutional layers and 3 fully connected layers. The convolutional layers are designed to learn local features from input images, while the fully connected layers are used for high-level feature extraction and classification. VGG-19 has a uniform architecture with small convolutional filters (3x3) and max pooling layers (2x2) applied sequentially, followed by fully connected layers. It is known for its deep architecture with a large number of parameters, which makes it suitable for tasks that require capturing fine-grained features in images.  One of the notable contributions of VGG-19 is its simplicity and the ability to train deep CNNs with relatively small filter sizes. It has been widely used as a benchmark architecture for image classification and feature extraction tasks in computer vision research and applications. VGG-19 has achieved top performance on various image recognition tasks, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and is often used as a base architecture for transfer learning, where pre-trained VGG-19 models are used as a starting point for fine-tuning on specific tasks with smaller datasets.  Top of Form  Bottom of Form |
|  |
| *Figure 3: VGG-19 Architecture* |
| **4.3.8 TENSORFLOW 2 AND KERAS** |
| TensorFlow 2 is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:   * Efficiently executing low-level tensor operations on CPU, GPU, or TPU. * Computing the gradient of arbitrary differentiable expressions. * Scaling computation to many devices, such as clusters of hundreds of GPUs. * Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.   Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.  Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device. |
| **4.3.9 JUPYTER NOTEBOOK** |
| Jupyter Notebook (formerly IPython Notebook) is a web-based interactive computational environment for creating notebook documents. Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and  rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension. |
| **4.3.10 NVIDIA CUDA** |
| **CUDA** (or **Compute Unified Device Architecture**) is a parallel computing platform and application programming interface (API) that allows software to use certain types of graphics processing units (GPUs) for general purpose processing, an approach called general-purpose computing on GPUs (GPGPU). CUDA is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels.  CUDA is designed to work with programming languages such as C, C++, and Fortran. This accessibility makes it easier for specialists in parallel programming to use GPU resources, in contrast to prior APIs like Direct3D and OpenGL, which required advanced skills in graphics programming. CUDA-powered GPUs also support programming frameworks such as OpenMP, OpenACC and OpenCL and HIP by compiling such code to CUDA. |
| **4.3.11 CANVA** |
| Canva is an Australian graphic design platform, used to create social media graphics, presentations, posters, documents and other visual content The app includes templates for users to use. The platform is free to use and offers paid subscriptions such as Canva Pro and Canva for Enterprise for additional functionality. In 2021, Canva launched a video editing tool. Users can also pay for physical products to be printed and shipped. It has announced it intends to compete with Google and Microsoft in the office software category, with website and whiteboard products. |

**CHAPTER 5**

**DESIGN**

|  |
| --- |
| **5.1 SYSTEM DESIGN** |
| System design is the process of defining the architecture, modules, and data for a system to satisfy specified requirements. It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate. The purpose of design phase is to plan a solution for problem specified by the requirements. System design aims to identify the modules that should be in the system, the specification of those modules and how they interact with each other to produce the result. The goal of the design process is to produce a model for or representation of a system can be used later to build. The produced model is called design of the system. |
|  |
|  |