# Cartoon Image Generation using ControlNet and Stable Diffusion

## ABSTRACT

In the evolving landscape of artificial intelligence and computer vision, transforming real-world images into stylized representations like cartoons has become a compelling area of research. This project presents an image-to-image generation pipeline using ControlNet integrated with Stable Diffusion, optimized for cartoonizing input images. By leveraging Canny edge detection as a control condition, the model captures essential structural outlines, guiding the Stable Diffusion model to generate visually appealing cartoon versions of real photographs. This implementation is executed on Google Colab with high efficiency using GPU acceleration. The ability to produce personalized, high-quality cartoon outputs from real images not only showcases the power of diffusion models but also opens doors for creative media production, digital avatars, and interactive entertainment.

## INTRODUCTION

Image stylization is a branch of image processing where real-world images are transformed into artistic or abstract representations. Among these, cartoonization—a process of converting a photographic image into a cartoon-like version—has garnered substantial attention in fields such as entertainment, gaming, education, and social media content creation. Traditional approaches for cartoonization involve complex filtering, hand-crafted feature extraction, or rule-based systems that often lack generalizability and visual richness.  
  
With the advent of deep learning, especially diffusion models, the paradigm shifted toward data-driven generative approaches. Diffusion models like Stable Diffusion have revolutionized text-to-image generation. These models learn to reverse a noise process to synthesize new images from textual descriptions. However, image-to-image generation using diffusion models was a relatively unexplored domain until frameworks like ControlNet emerged.  
  
ControlNet enhances the controllability of Stable Diffusion by conditioning the image generation on structured inputs like edges, depth, or human poses. In this project, we utilize the Canny edge detection technique to guide the diffusion process, helping the model understand the boundaries and contours in the input image. This edge map serves as a control signal for ControlNet, ensuring that the generated cartoon image retains the structure of the original photo while adhering to the stylistic prompt.  
  
Our pipeline is implemented using the diffusers library by Hugging Face in Google Colab. Users can upload an image, which is resized, processed into a Canny edge map, and then fed into the pipeline with a textual prompt like “cartoon version of the image.” The model outputs a cartoon-style image with preserved shapes and cartoon-like textures.  
  
This approach eliminates the need for large training datasets or fine-tuning and can be used on any input photo. It brings out the versatility and robustness of generative AI when combined with intelligent conditioning techniques like ControlNet. In this document, we elaborate on the scope, algorithm, implementation, and potential applications of our image-to-cartoon generation model.

## SCOPE AND MOTIVATION

The motivation behind this project arises from the growing demand for personalized and creative image transformation tools. Applications like profile picture stylizers, avatar creators, and animated content generators require fast, reliable, and visually appealing cartoonization methods. Traditional techniques often fall short in maintaining high fidelity or require manual intervention. This calls for an intelligent, fully automated cartoonization model that balances structure preservation and artistic abstraction.  
  
The scope of this work includes:  
- Developing an efficient pipeline for cartoonizing images using open-source tools.  
- Leveraging Canny edge detection to guide the image generation for structure preservation.  
- Using Stable Diffusion and ControlNet to control the stylization and maintain quality.  
- Deploying the solution on a cloud platform like Google Colab to make it accessible and GPU-accelerated.  
- Ensuring flexibility in the prompt-based style description for customization.  
  
This project fits well into areas like virtual avatars, digital marketing, media content creation, and educational tools. The modular design allows it to be extended to other styles like sketch, anime, or oil painting by simply modifying the prompt or control technique. The intersection of visual AI and creativity continues to inspire such innovations.

## DESCRIPTION

Tools and Libraries Used:  
- HuggingFace diffusers: For loading and running Stable Diffusion pipelines.  
- ControlNet: For conditioning the generation with structural input (Canny edges).  
- Google Colab: For ease of use and GPU acceleration.  
- OpenCV: For generating Canny edge maps.  
- PIL (Python Imaging Library): For image processing and visualization.  
  
Pipeline Breakdown:  
  
1. User Input:  
 The user uploads an image via Colab file uploader. The image is resized to 512x512, the ideal input size for the model.  
  
2. Edge Detection:  
 A Canny edge detector is applied using OpenCV to extract the outline of the image. This serves as the conditioning input.  
  
3. Model Loading:  
 ControlNet (Canny version) is loaded and attached to the Stable Diffusion base model (v1.5). Both are loaded in half precision (fp16) for memory efficiency.  
  
4. Prompt-Based Generation:  
 A prompt like “Create a cartoon version of the image” is provided. The model processes the original and edge images to generate the cartoon output.  
  
5. Saving the Output:  
 The result is saved with a filename that prepends "Cartoon\_" to the original image name. The image is displayed using IPython.display.

## ALGORITHM OR PSEUDOCODE

High-Level Pseudocode:  
  
# Load ControlNet and Stable Diffusion model  
load\_model():  
 controlnet = load\_controlnet("sd-controlnet-canny")  
 base\_model = load\_sd("runwayml/stable-diffusion-v1-5")  
 attach(controlnet, base\_model)  
 move\_to\_cuda()  
  
# Upload image and process  
upload\_image():  
 input = user\_upload()  
 resized\_img = resize(input, 512, 512)  
 edge\_img = apply\_canny(resized\_img)  
  
# Generate cartoon image  
generate\_cartoon(input\_image, edge\_image, prompt):  
 result = model.generate(prompt=prompt,  
 image=input\_image,  
 control\_image=edge\_image)  
 return result  
  
# Save the image  
save\_image(result, input\_name):  
 name = "Cartoon\_" + input\_name + ".png"  
 save(result, name)

## IMPLEMENTATION

The complete implementation is performed in Python in a Colab environment. Here's a snapshot:  
  
- Libraries installed using pip install diffusers accelerate transformers xformers.  
- Model weights fetched from HuggingFace.  
- Input image uploaded using google.colab.files.  
- Edge image generated using OpenCV.  
- Output image is saved and shown using PIL and IPython.display.  
  
Code Snippet Highlights:  
  
controlnet = ControlNetModel.from\_pretrained(...)  
pipe = StableDiffusionControlNetImg2ImgPipeline.from\_pretrained(...)  
uploaded = files.upload()  
original\_image = Image.open(input\_path).convert("RGB")  
canny\_image = get\_canny(original\_image)  
result = pipe(prompt=..., image=original\_image, control\_image=canny\_image)

## OUTPUT

Sample outputs show that:  
  
- Structural integrity of the original photo is maintained.  
- Stylization is smooth with strong outlines and simplified color patches.  
- Different prompts (e.g., "cartoon," "anime," "comic") give varied artistic effects.  
  
Output Example:  
  
Input Image ──► Canny Edge ──► Cartoonized Output

## APPLICATIONS

- Profile Picture Stylization: Generate custom cartoon avatars.  
- Entertainment: Digital caricatures and comic filters.  
- Education: Storyboard creation for children.  
- Marketing: Personalized branding material.  
- Gaming: Character avatars for non-realistic game design.

## CONCLUSION

This project demonstrates the power of diffusion models when combined with structure-preserving conditioning like ControlNet. The model delivers cartoonized outputs that maintain the shape of original photos while transforming their texture and style. The modular design of the pipeline allows customization through different prompts and can easily be extended to other artistic styles. The use of open-source tools and Google Colab ensures accessibility and cost-efficiency, making this solution suitable for real-world creative applications in both academia and industry.