combined

rounak Gera

August 2024

1 Introduction

The image inpainting is a process of reconstructing the missing regions of an image requiring strong understanding of image structure as well as semantic understanding of the image. Eg: If you have a image with some burned regions , you can correct it using image inpainting. It aims at realistic filling of missing parts of a image. It has a vast number of applications in image restoration, object removal and replacement etc.

It has multiple approaches like traditionally it was majorly done by using nearby neighboring pixels to estimate the value of current pixel. But with the upcoming of deep learning techniques it is completely revolutionised with more plausible , context aware and high resolution filling of missing regions. Earlier CNN's were used for such computer vision based problems but with the success of transformers in natural language processing it has also showed its ability to produce excellent results in computer vision tasks especially in problems where contextual awareness is needed like in image inpainting problems.

But it still faces two major issues which are small receptive field and non realistic and inappropriate filling of missing parts which are solved by two methods i.e. LAMA and MAT where LAMA provides an approach to increase the receptive field by using FFC(Fast Fourier Convolutions) based on converting the inputs to frequency domain and MAT proposed a transformer based method capable of generating high-resolution outputs for large masks.

In this paper we combine the methods of LAMA and MAT using a third combiner model based on UNET architecture and a denoiser model to remove noise generated during inpainting. We show an approach based on a mix of transfer learning and ensemble learning to achieve high efficiency without high computational requirements. But as it is known that LAMA and MAT both are quite heavy we use a step by step approach to preform inference of our ensemble model using a GPU of 16GB RAM .Our contributions can be summarized as

(i) Making an ensemble model which combines both LAMA and MAT along with making use of denoisers to improve the results.
(ii) Step by step approach to train the ensemble model in a GPU of 16 GB RAM.

2 Preliminaries

1. Ensembling is a technique of combining two or more models in order to get better results and improve the efficieny of model by combining the power of multiple models. It is primarily of two types i.e. bagging and boosting.
2. Bagging is a type of ensemble learning where multiple models are trained in parallel and average of their outputs is taken to return the final output.
3. Boosting is a type of ensemble learning where multiple models mostly weak learners are ran sequentially with each model learning from mistakes of previous models.
4. Transfer learning is a technique where pretrained model on one task is fine tuned to be used on any other custom task so as to avoid all the time and cost of training the model on custom case.
5. Denoising models are models which are used to obtain a clean image from a noisy image.
6. U-Net is a convolutional neural network(CNN) primarily developed for image segmentation consisting of a encoder, bottleneck and decoder.

3 Proposed Approach

Our methods solves the 16 GB GPU RAM constraint by a step by step process where the whole process is distributed into seprate parts to make the best use of GPU along with ensembling the LAMA and MAT model and also additionally utilizing the denoising models to improve the performance and quality of outputs.

3.1 Ensembling

For ensembling both the models there were two alternative i.e. boosting or bagging but since bagging would have involved parallel working of both models so it would have easily broken our memory constraint while boosting would have fitted into our memory constraint but the fact that there is n't any best way to measure the error which could satisfy our purpose , So to choose a middle path , it was decided to use **stacking** that first the input will be processed by first model and then after clearing the memory properly the same input file will be processed by second model and the memory will be again cleared and later on the outputs of both models will be processed and combined by denoising and combiner models.

3.2 Transfer Learning

So although the models were now in our memory constraint but still there was one problem that the training time was too much even when it was decided that the models will be trained on only a 25 percent subset of Places365 standard train dataset which consist of 1.8 million images but still the training time was too much to be done on google colab since the models are quite deep and complex and also the number of models are more so to solve the problem it was decided to opt for transfer learning and it proved to be a great. Where the pretrained models of the original authors of models(LAMA and MAT) were used to avoid training which also helped in improving the performance. Although for denoising models and UNET based ensemble model were trained due to unavailabilty of trained weights for similar task , additionally these models were much lighter than LAMA and MAT so it was decided to train and they were trained on 7300 images consisting of 20 images from each scene category from Places365 standard train dataset.

3.3 Combiner model

Finally a model was made which could combine both the models results which was similar to a classic U-Net without residual connections consisting of three downsample blocks and three upsample blocks along with ReLU as activation layer.But the model seemed to make very little progress while training and model seemed to overfit which was easily avoided by changing the activation functions from ReLU to LeakyReLU. But still the outputs of the ensemble models were not upto mark and it was observed that there was some noise on output produced by LAMA model so it was decided to use a denoiser model on LAMA's output and then the ensemble model was run on its output and MAT's output but this time the final output contained some noisy patches which was quite easily solved by using another denoising model of same architecture. Then the final output was ready.

3.4 Denoising Model

Denosing models played a pivotal role in improving our final output, at first instance the denoising model helped to improve the output produced by LAMA model which was fed to our ensemble / combiner model and at second instance denoising model helped to correct the random blocks of noise in output of combiner model and then the final output was made. The denoising model is a convolutional neural network(CNN) consisting of encoder and decoder block where the encoder block consisted of three convolutional layers along with LeakyReLU for downsampling and decoder consisted of three convtranspose with LeakyReLU for upsampling and finally a sigmoid activation function at end.

3.5 Workflow

First of all the LAMA model was used to and it was loaded with the pre-trained weights and inference was ran over validation set of Places365 dataset which was processed according to the requirements of model then the resulting images were stored for later use in combiner model, also the masks were also stored to ensure uniformity across outputs of both models. Then the same process was

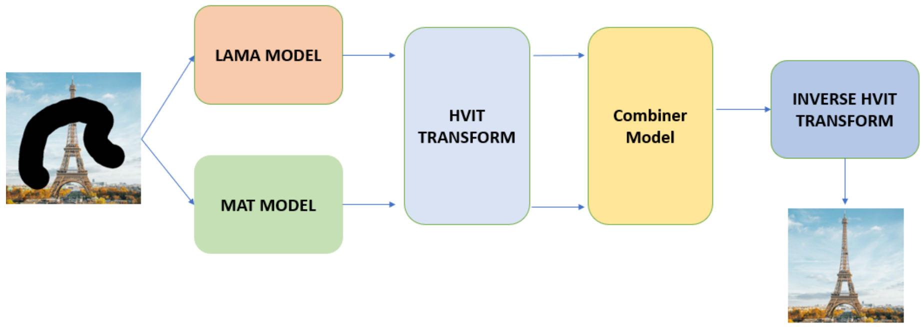

Figure 1: The figure illustrates the workflow of our technique, starting with the masked image which is fed in parallel to both LAMA and MAT model, then the both output images are converted to HVIT from RGB, then both the converted images are fed to combiner model and finally the resulted image which is in HVIT color space is converted again to RGB space by using inverse HVIT transform function to obtain the final output image

repeated for the pre-trained MAT model and its results were also stored and it is important to note that the masks were same. Then the output of LAMA model were passed on to our previously trained denoising model and then the output of MAT and denoising models were fed to combiner model then the resulting output was again passed on to our other previously trained denoising model to generate our best outputs.

One small issue was that the saved masks were being loaded as RGB images and had values in range of 0 to 255 which was inappropriate for our purpose of masking the images which was solved using threshold function of opencv.

Equation :-

Y = Final denoiser (Combiner Model(Lama Denoiser (LAMA (X)), (MAT (X)))

3.6 Loss Function

For purpose of computing loss for training our ensemble model four different loss functions were used to get the best measure of loss. Firstly Style loss was used for analysing texture, patterns along with capturing complex stylistic features far beyond simple color distributions of images. Next Edge loss was used which help to ensure sharpness and clear edges which further help in maintaining the structural integrity of objects in image. Next Perceptual loss based on pre-trained VGG model , it was used for comparing feature similarity ensuring structural and semantic similarities and at last MSE loss was used to measure differences at pixel level.

$$\mathcal{L}_{total} = \lambda_{style}\mathcal{L}_{style} + \lambda_{edge}\mathcal{L}_{edge} + \lambda_{perceptual}\mathcal{L}_{perceptual} + \lambda_{MSE}\mathcal{L}_{MSE}$$

For purpose of training our denoising models only MSE loss was used due to simplicity, effectiveness and proven record of good results for denoising task as it measures difference at pixel level and in case of task like denosing it is much needed.

$$\mathcal{L}_{DenoisingModel} = \mathcal{L}_{MSE}$$

3.7 Data Preparation

For training of combiner model and denosing models we used a subset of Places365 standard train set where we took 20 images from each scene category totaling to 7300 images.

For evalutaion we use the validation set of Places365 dataset consisting of 36,500 images, then we remove all the images which don't have all the three color channels and also we resize all the images to 256X256 size to ensure uniformity in images. Also, we take care of the fact that even masks are read as color images and have values between 0 to 255. So by using the threshold function we convert all the values to 0 and 1's. Then we convert the images to tensors using a transform function.

3.8 Step by step procedure

1. Get the data of Places365 validation set and preprocess it as explained in above sections
2. Generate masks using the technique introduced in LAMA paper and store them for later use and make masked images .
3. Perform inference of pre-trained LAMA model on our masked images and store the results .
4. Repeat the step 3 for pre-trained MAT model.
5. Now apply the LAMA denosing model on the resultant images of LAMA model.
6. Feed the converted images and MAT images to Combiner model .
7. Now apply final denoising model on the resulted images to obtain the final and improved output images.

4 Experimentation

so here our all experiments ans results

5 Summary

In recent times various techniques in field of image inpainting are introduced for solving problems like contextual ambiguity, visual inconsistency, low resolution, low receptive field and many other. With most of the techniques using transformers and diffusion as base. While quite innovative techniques are also introduced including using graph based approach , using signal processing techniques for image inpainting. Also some techniques have introduced multi-modals or multi-task generative models which can do a wide set of tasks like, object removal, text guided image [?], context aware, and shape guided inpainting.

While GAN's and Variational Auto Encoders (VAE) which were used intensively for tasks like image inpainting have seen a downfall in their usage after the introduction of transformers and diffusion. Where transformers excel in semantic understanding of images and diffusion excel in generating high quality and realistic images.

For text guided inpainting most of the techniques use learnable prompts which guide the model to generate content as per instructions. Also some techniques have used coarse to fine approach where a coarse network fill the missing regions which is later refined by a fine network to produce high quality results. Some techniques have used edge detection which provide a rough structure of image to help the model in generating contextually stable results.

Most of the techniques use perceptual loss, style loss, total variational loss and adversarial loss. Commonly used evaluation metrics include SSIM(Structural similarity index measure), FID(Fréchet inception distance) , PSNR(peak signal-to-noise ratio) and LPIPS(Learned Perceptual Image Patch Similarity). Commonly used includes Places365 and CelebA but there are very less models trained on large scale datasets like LAION-5B.