Astronomical Image colorization and up-scaling with generative adversarial networks

1st Shreyas Kalvankar *
Dept. of Computer Engineering
K. K. Wagh Institute of Engineering Education and Research
Nashik, India
shreyas.kalvankar@kkwagh.edu.in

2nd Snehal Kamalapur*
Dept. of Computer Engineering
K. K. Wagh Institute of Engineering Education and Research
Nashik, India
smkamalapur@kkwagh.edu.in

3rd Hrushikesh Pandit*
Dept. of Computer Engineering
K. K. Wagh Institute of Engineering Education and Research
Nashik, India
hrushikesh.pandit@kkwagh.edu.in

4th Pranav Parwate*
Dept. of Computer Engineering
K. K. Wagh Institute of Engineering Education and Research
Nashik, India
pranav.parwate@kkwagh.edu.in

5th Atharva Patil*
Dept. of Computer Engineering
K. K. Wagh Institute of Engineering Education and Research
Nashik, India
atharva.patil@kkwagh.edu.in

Abstract—Automatic colorization of images without human intervention has been a subject of interest in the machine learning community for a brief period of time. Assigning color to an image is a highly ill-posed problem because of its innate nature of possessing very high degrees of freedom; given an image, there is often no single color-combination that is correct. Besides colorization, another problem in reconstruction of images is Single Image Super Resolution, which aims at transforming low resolution images to a higher resolution. This research aims to provide an automated approach for the problem by focusing on a very specific domain of images, namely astronomical images, and process them using Generative Adversarial Networks (GANs). We explore the usage of various models in two different color spaces, RGB and L*a*b. We use transferred learning owing to a small data set, using pre-trained ResNet-18 as a backbone, i.e. encoder for the U-net and fine-tune it further. The model produces visually appealing images which hallucinate high resolution, colorized data in these results which does not exist in the original image. We present our results by evaluating the GANs quantitatively using distance metrics such as L1 distance and L2 distance in each of the color spaces across all channels to provide a comparative analysis. We use Fréchet inception distance (FID) to compare the distribution of the generated images with the distribution of the real image to assess the model's performance.

Index Terms—Neural Networks, Generative Models, Image Colorization, Image Super-resolution

I. INTRODUCTION

Automated colorization of gray scale images has been subjected to much research within the computer vision and machine learning communities. Beyond simply being fascinating from an aesthetic and artificial intelligence perspective, such capability has broad practical applications. It is an area of research that possesses great potentials in applications: from black and white photo reconstruction, image augmentation, video restoration to image enhancement for improved interpretability. Image downscaling is an innately lossy process. The principal objective of super resolution imaging is to reconstruct a low resolution image into a high resolution one based on a set of low-resolution images to rectify the limitations that existed while the procurement of the original low-resolution images. This is to insure better visualization and recognition for either scientific or non-scientific purposes. Even if an upscaling algorithm is particularly good, there will always be some amount of high frequency data lost from a downscale-upscale function performed on the image. Ultimately, even the best upscaling algorithms are unable to effectively reconstruct data that does not exist. Traditional methods for image upsampling rely on low-information, smooth interpolation between known pixels. Such methods can be treated as a convolution with a kernel encoding no information about the original image. A solution to the problem is by using Generative Adversarial Networks (GANs) to hallucinate high frequency data in a super scaled image that does not exist in the smaller image. Even though they increase the resolution of an image, they fail to produce the clarity desired in the super-resolution task. By using the above mentioned method, not a perfect reconstruction can be obtained albeit instead a rather plausible guess can be made at what the lost data might be, constrained to reality by a loss function penalizing deviations from the ground truth image.

A huge number of raw images are present unprocessed and unnoticed in the Hubble Legacy Archives. These raw

(a) Ground Truth
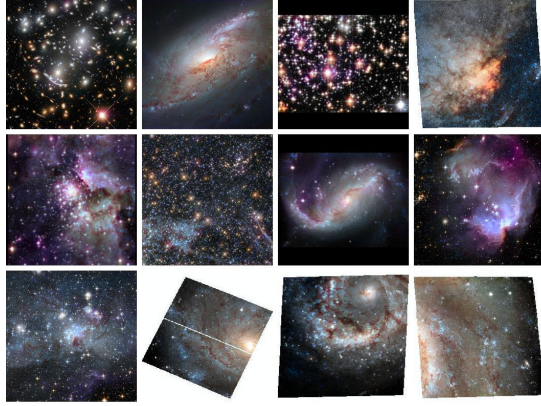
(b) ResNet18 Full trained L*a*b

Fig. 1: Results of fine-tuned ResNet18 in L*a*b colorspace compared with ground truth images

images are typically black and white, low-resolution and unfit to be shared with the world. It takes huge amounts of hours to process them. This processing is necessary because it's difficult for astronomers to distinguish objects from the raw images. Random and synthetic noise from the sensors in the telescope, changing optical characteristics in the system and noise from other bodies in the universe all make the processing further necessary. Furthermore, for the process of highlighting small features that ordinarily wouldn't be able to be picked out against noise of the image, we need colorization. The processing of the images is so time consuming that the images are rarely seen by human eyes. The problem is only likely to get worse. Not only is new data being continuously produced by Hubble Telescope, but new telescopes are soon to come online. A simplification of image processing by using artificial image colorization and super-resolution can be done in an automated fashion to make it easier for astronomers to visually identify and analyze objects in Hubble dataset.

II. LITERATURE REVIEW

A. Image Colorization

Hint Based Colorization

Levin et al. (2004) proposed using colorization hints

from the user in a quadratic cost function which imposed that neighboring pixels in space-time with similar intensities should have similar colours. This was a simple but effective method but only had hints which were provided in form of imprecise colored scribbles on the grayscale input image. But with no additional information about the image, the method was able to efficiently generate high quality colorizations. Huang et al. (2005) addressed the color bleeding issue faced in this approach and solved it using adaptive edge detection. Yatziv and Sapiro (2006) used luminescence based weighting for hints to boost efficiency. Qu et al. (2006) extended the original cost function to apply color continuity over similar textures along with intensities.

Deep Colorization

Owing to recent advances, the Convolutional Neural Networks are a de facto standard for solving image classification problems and their popularity continues to rise with continual improvements. CNNs are peculiar in their ability to learn and differentiate colors, patterns and shapes within an image and their ability to associate them with different classes.

Cheng et al. (2016) proposed a per pixel training for neural networks using DAISY (Tola et al.; 2008), and semantic (Long et al.; 2015) features to predict the chrominance value for each pixel, that used bilateral filtering to smooth out accidental image artifacts. With a large enough dataset, this method proved to be superior to the example based techniques even with a simple Euclidean loss function against the ground truth values.

Finally, Dahl (2016) successfully implemented a system to automatically colorize black & white images using several ImageNet-trained layers from VGG-16 (Simonyan and Zisserman; 2015) and integrating them with auto-encoders that contained residual connections. These residual connections merged the outputs produced by the encoding VGG16 layers and the decoding portion of the network in the later stages. He et al. (2015) showed that deeper neural networks can be trained by reformulating layers to learn residual function with reference to layer inputs. Using this *Residual Connections*, He et al. (2015) created the *ResNets* that went as deep as 152 layers and won the 2015 ImageNet Challenge.

Generative Adversarial Networks

Goodfellow et al. (2014) introduced the adversarial framework that provides an approach to training a neural network which uses the generative distribution of $p_g(x)$ over the input data x.

Since it's inception in 2015, many extended works of GAN have been proposed over years including DCGAN (Radford et al.; 2016), Conditional-GAN (Mirza and Osindero; 2014), iGAN (Zhu et al.; 2018), Pix2Pix (Isola et al.; 2018).

Radford et al. (2016) applied the adversarial framework for training convolutional neural networks as generative models for images, demonstrating the viability of *deep*

convolutional generative adversarial networks.

DCGAN is the standard architecture to generate images from random noise. Instead of generating images from random noise, Conditional-GAN (Mirza and Osindero; 2014) uses a condition to generate output image. For e.g. a grayscale image is the condition for colorization of image. Pix2Pix (Isola et al.; 2018) is a Conditional-GAN with images as the conditions. The network can learn a mapping from input image to output image and also learn a separate loss function to train this mapping. Pix2Pix is considered to be the state of the art architecture for image-image translation problems like colorization.

B. Image Upscaling

Frequency-domain-based SR image approach

TSAI (1984) proposed the frequency domain SR method, where SR computation was considered for the noise free low resolution images. They transformed the low resolution images into Discrete Fourier transform (DFT) and further combined it as per the relationship between the aliased DFT coefficient of the observed low resolution image and that of unknown high resolution image. Then the output is transformed back into the spatial domain where a higher resolution is now achieved.

While Frequency-domain-based SR extrapolates high frequeny information from the low resolution images and is thus useful, however they fall short in real world applications.

The interpolation based SR image approach

The interpolation-based SR approach constructs a high resolution image by casting all the low resolution images to the reference image and then combining all the information available from every image available. The method consists of the following three stages (i) the registration stage for aligning the low-resolution input images, (ii) the interpolation stage for producing a higher-resolution image, and (iii) the deblurring stage which enhances the reconstructed high-resolution image produced in the step (ii).

However, as each low resolution image adds a few new details before finally deblurring them, this method cannot be used if only a single reference image is available.

Regularization-based SR image approach

Most known Bayesian-based SR approaches are maximum likelihood (ML) estimation approach and maximum a posterior (MAP) estimation approach.

While Tom and Katsaggelos (1996) proposed the first ML estimation based SR approach with the aim to find the ML estimation of high resolution image, some proposed a MAP estimation approach. MAP SR tries to takes into consideration the prior image model to reflect the expectation of the unknown high resolution image.

Super Resolution - Generative Adversarial Networks (SR-GAN)

The Genrative Adversarial Network (Goodfellow et al.;

2014), has two neural networks, the Generator and the Discriminator. These networks compete with each other in a zero-sum game. Ledig et al. (2017) introduced SRGAN in 2017, which used a SRResNet to upscale images with an upscaling factor of 4x. SRGAN is currently the state of the art on public benchmark datasets.

III. METHODOLOGY

A. Data collection

We started by scraping data off the Hubble Legacy Archive. The scrapper tool[1] which was used, courtesy of Peh and Marshland (2019)[2], scraped off hundreds of thousands of colorized images the archive has available. The Hubble Legacy archive is slow and produces grainy images with lots of noise and majority are unprocessed. A filter for M101 (Messier 101) galaxy rendered more than 80 thousand images with a 1 degree difference between consecutive right ascension. The data acquired was large and there was no particularly efficient way to clean it without human investment. Cleaning tens of thousands of images by handpicking noiseless and well colored images is time consuming. For training the SRGAN, high resolution and well colored images were needed.

Consequently, we scraped the Hubble Heritage project instead. The Hubble Heritage project releases the highest-quality astronomical images. They are all stitched together, colorized and processed to eliminate noise. Hubble Heritage then selects the best, most striking of these for public release. However, there were only $\sim$ 150 of these images that are actually useful. We scraped images from the main Hubble website as well so as to increase the amount of data we had. This provided another $\sim$ 1000 images approx. Limited by our computational resources, it was decided that the images will be in the dimensions of 256×256 pixels and contains 3 channels namely Red, Green, Blue (RGB).

B. Image Color Space

An RGB image is essentially a rank 3 tensor of height, width and color where the last axis contains the color data of our image. The data is represented in RGB color space which has 3 numbers for every pixel indicating the amount of *Red, Green, and Blue* values the pixel has. In **L*a*b** color
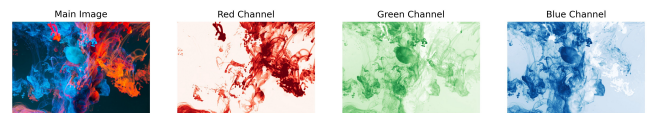

Fig. 2: Reg, Green and Blue channels of an image (Main image by Lucas Benjamin)

space, we have three numbers for each pixel but these numbers

have different meanings. The first channel, L, encodes the **Lightness** of each pixel and when we visualize this channel it appears as a black and white image. The ***a and *b** channels
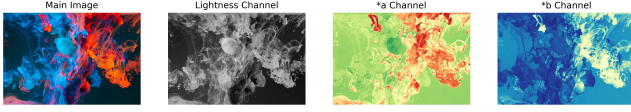

Fig. 3: Lightness, *a and *b channels of the L*a*b colorspace

encode how much **green-red** and **yellow-blue** each pixel is, respectively. Guadarrama et al. (2017); Isola et al. (2018) propose the use of L*a*b color space instead of RGB to train the models. Intuitively, to the train a model for colorization, we should give it a grayscale image and we hope that it colors it. In the L*a*b colorspace, the model is fed the L channel, which is essentially a grayscale image, and we perform computations to predict the other two channels (*a and *b). After the predictions, we concatenate the channels and get a colorful image. In case of RGB, there is a need to explicitly convert the three channels down to 1 to make it a grayscale image and then feed it to the network hoping that it predicts three numbers per pixel. This is an unstable task due to sheer increase in volume of combinations from two numbers to three.

We train models using both color spaces and compare their performance.

C. Transferred learning and model tweaking

Isola et al. (2018) proposed a general solution to many image-to-image translation tasks. We propose a similar methodology as proposed in the paper with a few minor tweaks, so as to significantly reduce the amount of training data needed and minimize the training time to achieve similar results. Initially, we propose the use of a U-net for the generator. A pre-trained ResNet-18 network was used as the *encoder* for the U-net with half of it's layers clipped off so as to get feature abstractions of the intermediate layers. Ledig et al. (2017) showed that training the generator separately in a supervised, deterministic manner helps it generalize the mapping from the input space to outputs. The idea solves the problem of *"The blind leading the blind"* that is persistent in most image-to-image translation tasks to date.

We pre-train our generator independently using the imagenet weights over the Common Objects in Context (COCO) dataset. Unlike adversarial training, this phase was supervised and the loss function used was *mean absolute error* or the *L1 Norm* from the target image. Though trained deterministically, the problem of rectifying incorrect predictions still persists due to constraints over the convergence of loss metric. To combat this, this trained generator was trained once again in an adversarial fashion to help generalize it further. We hypothesize that re-training in an adversarial fashion will further rectify the subtle color differences that *mae* couldn't solve.

A pre-trained ResNet-50 with imagenet weights was used as the discriminator with last few layers clipped off. The discriminative network used is something called a "Patch Discriminator" proposed by Isola et al. (2018). In a *vanilla* discriminator proposed by Goodfellow et al. (2014), the network produces a scalar output, representing the probability that the data x belongs to the input distribution and not the generator distribution p_g. Isola et al. (2018) proposed a modification to the discriminative network so as to produce, instead of a single scalar, a vector of probabilities representing different localities of the input distribution (image) x. For instance, a discriminator producing a 50×50 vector represents the probabilities every receptive field that is covered by the output vector. Thus, we can localize the corrections in the image that the generator should make.

Finally, the trained generator and trained discriminator are fine-tune to fit on our data which is rather small in size compared to the previous datasets. These networks are trained in an adversarial fashion using the conditional GAN objective function (Isola et al.; 2018) with some noise introduced as the L1 norm of generated image tensor to the target image tensor. The reason behind this is to train the generator using an adversarial component while trying to minimize the Manhattan distance between the generator output and target vector space.

D. Experimental Setup

We use a mini-batch gradient descent with a batch size of 10, 16 and 32 for different iterations. We use Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The generator and discriminator has a learning rate of $2e - 4$ which remains constant throughout the training. We train the model with different epochs ranging between 20,50 and 100, saving the best model weights determined by L1 norm between the output of the generator and the target image. We use early stopping with a patience value of 10. The image size is 64×64. Our implementation is using Python, numpy, Tensorflow and tf-keras. It takes about 24 hours to complete training over the COCO dataset and about 12 to 13 hours to fit the model on given astronomical data on a NVIDIA Tesla K80 GPU.

IV. RESULTS AND DISCUSSIONS

In the following section, we briefly compare and summarize the results of the implemented architectures and evaluate their performance. The evaluation is done qualitatively as well as quantitatively. We compare the performance of each model using L1 and L2 norm distance between the predictions and targets. Though unreliable, this method provides us with a somewhat decent ground to perform a comparative study and evaluate the reliability of such metrics on GAN evaluation compared to qualitative, visual evaluation. We also measure the performance of the GANs by using Fréchet inception distance (FID) as an attempt to quantify how close the produced images are to the original ones.

To evaluate the model performance by virtue of convergence of the objective function, we plot the generator loss throughout the training on two different networks.

Figure [4a] and [4b] shows that the generator converges quite
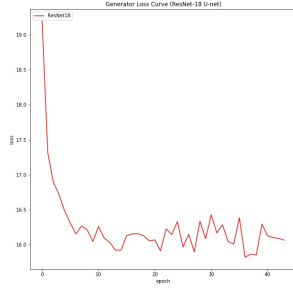
(a) ResNet18 U-net Generator
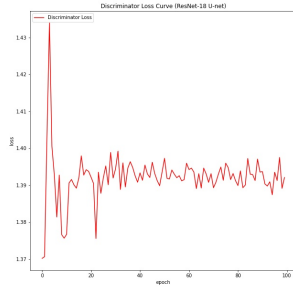
(b) ResNet50 discriminator loss

Fig. 4: Curve plots for ResNet18 U-net & ResNet50 discriminator

nicely. The discriminator on the other hand oscillates because of the convergence that the generator shows. GAN losses are pretty non-intuitive but we can draw some observations from the loss curves. It seems that the pattern of oscillation and convergence repeats when networks are trained in an adversarial fashion. It is observed that the ResNet U-net with a pre-trained ResNet-18 for its backbone converges decently in the beginning but later shows spikes when nearing the end of training loop. It seems that the loss convergence doesn't signify whether the model is predicting expected results. The loss function just converges to the minimum value that the cost function descends to, permitted by the learning rate. It would normally signify that the GAN has found some optimum point in the vector space that is at the lowest potential and cant decrease any further. It essentially means that the GAN has leaned enough. Due to the large number of dimensions, owing to the high amount of trainable variables, such combinations, where the function converges, can be huge in volume. Thus, these numbers don't provide any better understanding of the bias or variance the model is facing. Also, we discover that if the loss hasn't converged well, it doesn't necessarily mean that the model hasn't learned anything. On visual inspection, the generated results show similar results to the ground truth, even with high generator losses. This might be due to presence of a content loss parameter in the loss function where we try to minimize the function by minimizing the L1 norm between

the generator predictions and target.

The discriminator shows an increase in the objective loss function in the initial epochs and then settles down in the later phase around an oscillation point. This shows that the discriminator is converging to some permanent number or rather, oscillating around it. We assume this point to be a point of stability between the two networks as the networks are in a constant adversarial battle, meaning if one performs better, the other is bound to perform worse.

A. Image Colorization

We present a comparative study of the following models: Basic U-net generator with a custom, residual VGG16 discriminator, hereafter referred to as Basic U-net. A modified U-net with pre-trained ResNet18 as its backbone. We evaluate this model by training it in RGB color space to predict 3 numbers for every pixel and in L*a*b color space where the model predicts the a and b channel alone. We then further train this model to fine-tune it to the task of colorizing astronomical images. From figure 5 we observe that on this particular
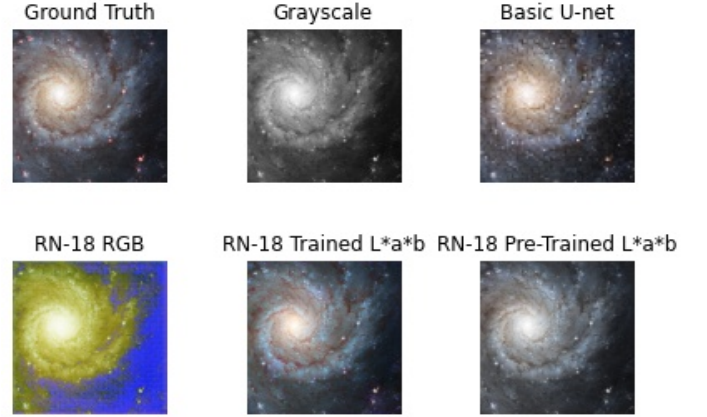

Fig. 5: Results of the colorization study. The first two images in the top row are the ground truth and input image respectively. The next images belong to the output of following, serially: Basic U-net, ResNet18 in RGB colorspace, ResNet18 fine-tuned in L*a*b colorspace, ResNet18 pre-trained U-net in L*a*b colorspace

example, the Basic U-net performs better at predicting results that closely map to the ground truth but the fine-tuned ResNet-18 shows promising results over a larger set of inputs. It can also be observed that the Basic U-net architecture does a decent job of faking the sharpness in the original image and that makes the image appear more realistic as compared to rather blurry outputs from the other networks.

We also observe that the model trained in the RGB color space performs poorly at predicting values of the three color channels and that results in dominance of one channel (blue in this case) over others. This causes the output, even though reconstructed quite accurately, to have a varied color pattern with high emphasis on one color channel. This might be because of presence of deep layers which cause gradients of

certain colors to diminish over time, causing the model to be strongly biased towards the blue color in this case. An increase in the volume of training data and some random pixel shuffling in forward propagation might solve this problem.

The pre-trained ResNet18 U-net performs decently with the weights gathered by training it over the COCO dataset. The model still lacks the specific coloring intuition in astronomical images and plainly colors specific parts of the images in light colors, leaving majority of the image unaltered. This causes the images to have a slight gray tinge.

Figure 7 shows how the other models perform in different color spaces. We can observe that the Basic U-net model performs good at predicting the outputs but doesn't predict the brightness level of pixels quite accurately. The model seems to be overfitting on the dataset and suffers a high variance on output because there are a lot of testing samples which demonstrate the poor performance of the Basic U-net model as can be seen in figure 6.
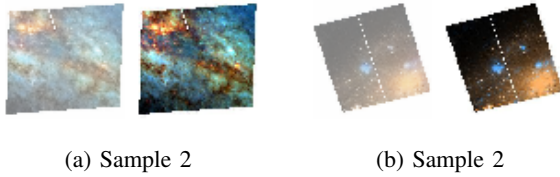

(a) Sample 2   (b) Sample 2

Fig. 6: Performance of Basic U-net over samples different from the dataset. The left part is the prediction, right part is the ground truth

Figure 1a shows the ground truth images and figure 1b shows the predictions of the ResNet18 full trained model in the L*a*b color space. This model seems to perform best, visually.

To quantitatively estimate the model performance, we measure the L1 and L2 distance between the predictions and the ground truth images in both RGB as well as L*a*b colorspace.

Model	Color Space	L1 Distance	L2 Distance
ResNet-18 (Pre-trained)	L*a*b	64.5409	2.77
ResNet-18 (Fine-tuned)	L*a*b	65.1119	2.62
ResNet-18 (Pre-trained)	RGB	125.5554	9.04
Vanilla/Basic U-net	RGB	77.3273	3.986

TABLE I: Colorization: Average per-pixel L1 & L2 distance between generated images and ground truth

Figure 5 shows that the RGB color space performs poorly on the task of colorization. In terms of the L1 distance, the best performance is achieved on the ResNet18 U-net with pre-trained weights. This goes to prove the unreliability of distance metrics in model performance evaluation. The model, although quantitatively, performs better than the fine-tuned model but in reality, fails to produce images that might be visually appealing. The L2 distance metric shows how the fine-tuned model might, in reality, be fitting slightly better over the data to predict correct color combination as its output. We

(a) Basic U-net

(b) ResNet18 pre-trained L*a*b
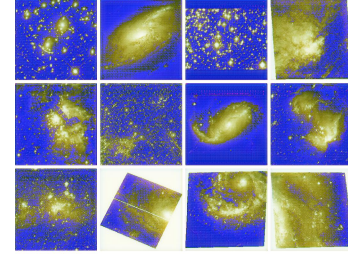
(c) ResNet18 pre-trained RGB

Fig. 7: Results of colorization models (from top to bottom): Basic U-net, ResNet18 U-net pre-trained in L*a*b color space, ResNet18 U-net pre-trained in RGB color space

still consider the results from the fine tuned model to be better than the other performing models and further investigate the per-channel predictions by the model.

Distance	Red	Green	Blue
L1 Norm	64.6078	38.5994	92.1283
L2 Norm	3.4531	1.0253	3.8046

TABLE II: Channel wise averages of ResNet-18 finetuned network

Table II shows the RGB channel averages of the outputs produced by the fine tuned model. It can be observed that the feature embeddings produced by the model in *a, *b color spaces maps to the green channel with the least error. This might be indicative that the model is performing poorly on images that have a high content of red-blue colors.

In table III, L1 distance, we can see the performance of fine-tuned model to be better in channel *a but poor in *b compared to the pre-trained model. The L2 distance metric rules out the possibility of the fine-tuned model performing better than the pre-trained model. In reality, the fine-tuned model is orders of

Model	L1 Distance		L2 Distance	
	a	b	a	b
ResNet-18 (Pre-trained)	0.00588	0.00501	0.01565	0.03208
ResNet-18 (Fine-tuned)	0.00257	0.00727	0.01721	0.03316

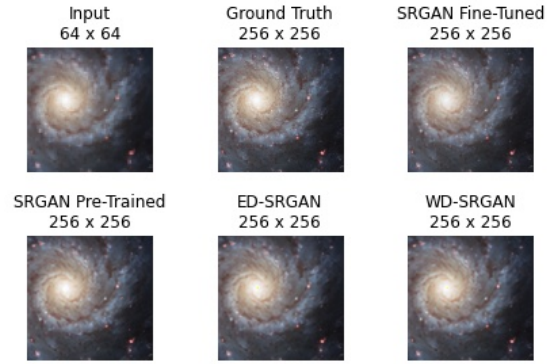TABLE III: L1 & L2 channel wise distances in L*a*b colorspace

Fig. 8: Results of the super-resolving models. The first two images in the top row are the input image and ground truth respectively. The next images belong to the output of following, serially: Fine-tuned SRGAN, pre-trained SRGAN, EDSR, WDSR

magnitude better at predicting output abstractions with the L channel as its input, thus contradicting the quantitative results.

Model	L1 Distance			L2 Distance		
	R	G	B	R	G	B
ResNet-18 (PT)	69.2401	36.5852	87.7973	3.2890	1.0439	3.9785
ResNet-18 (FT)	64.6078	38.5994	92.1283	3.4531	1.0253	3.8046

TABLE IV: L1 & L2 channel wise distances in RGB colorspace (PT refers to Pre-trained and FT refers to Fine-tuned)

We can observe the comparative results of ResNet18 pre-trained model and the fine-tuned model. It can be concluded from the observations that the fine-tuned model performs better at predicting the Red color channel but suffers in Green and Blue.

Model	FID
ResNet-18 (Pre-trained)	66.3568
ResNet-18 (Fine-tuned)	42.1593
U-net	152.7216

TABLE V: Colorization: Fréchet inception distance between model outputs and original images

We use Fréchet inception distance (FID) to evaluate the distribution of generated images with the distribution of real images. Table V helps us see that the fine-tuned U-net with ResNet-18 as it's backbone achieves the least FID score. This still shows that while this model is very adept at hallucinating images but is still not able to predict accurate color values.

B. Image Super Resolution

We implement the basic SR-GAN proposed by Ledig and train it to improve super-resolution task. We compare the trained model with pre-trained SRGAN model, EDSR-GAN proposed by Lim et al. (2017) and WDSR-GAN proposed by Yu et al. (2018).

Figure 8 shows how the networks perform on the task of predicting pixels while upscaling of the image. All the networks perform really well and its difficult to distinguish the outputs visually. It may be observed that the fine-tuned network produces images that look slightly better than the other counter-parts. The best performing network seems to be the WDSR-GAN with a very little error in predicting the output.

(a) Input Distribution
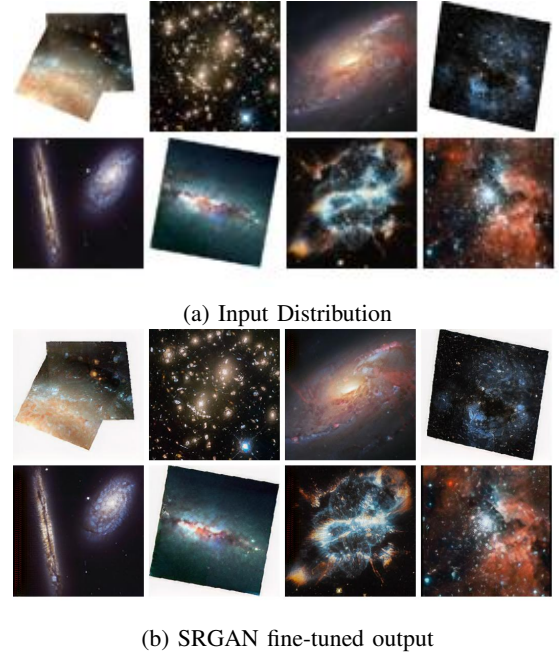

(b) SRGAN fine-tuned output

Fig. 9: Results of SRGAN. 9a shows the input data and 9b shows the corresponding output

It is evident that the model produces acceptable results on visual inspection. The main reason behind this might, again, be the random pixel shuffling between every upscaling pass.

As opposed to colorization, super-resolution needs a quantitative estimation to determine which model performs best among the give models.

Model	L1 Distance	L2 Distance
Ledig SRGAN (Fine-tuned)	87.1090	3.754
Ledig SRGAN (Pre-trained)	114.8043	5.953
ED-SRGAN	80.9414	3.684
WD-SRGAN	79.7262	3.627

TABLE VI: Super-Resolution: Per-pixel Average L1 & L2 distance between generated images and ground truth

Table VI shows the L1 and L2 distances of predicted results by each model with the ground truth image. It is observed that Ledig's SRGAN, after a bit of fine-tuning performs really well in comparison to the the pre-trained version. To further improve this, Lim et al. (2017) proposed an optimized version of SRGAN by removing the unnecessary modules in the conventional resnets and showed that Enhanced Deep Residual Networks performed better at upscaling task. When trained in an adversarial manner, the ED-SRGAN performs better than the traditional SRGAN. Yu et al. (2018) further improved the idea by increasing the widening factor ($\times 2$ and $\times 4$) to ($\times 6$ and $\times 9$). This Wide Activation Deep Super Resolution network further improved the performance for single image super-resolution. When we implement this in an adversarial manner, we achieve excellent results. It is evident from the results that the best performing network is the WDSR.

Model	FID
Ledig SRGAN (Fine-tuned)	75.5873
Ledig SRGAN (Pre-trained)	89.8443
ED-SRGAN	159.5770
WD-SRGAN	127.2920

TABLE VII: Super-resolution: Fréchet inception distance between model outputs and original images

Having said that, if we look at the FID score listed in table VII, we notice that the lowest FID score is given by the fine-tuned SRGAN (which is ideally what we expect), but this just proves our previous claim that GANs are particularly good at hallucinating things rather than predicting them.

V. Summary and Conclusion

The project mainly tackles the problem of colorizing astronomical images and super-resolving them for astronomical inspection. We explore various methodologies proposed till date that efficient colorize and super-scale images with results that are significantly closer to the ground truth distribution. We scrape the data from Hubble Legacy Archive, Hubble Main website and Hubble Heritage project and created a filtered and clean dataset of 4700 images. We split the data and use 500 images, roughly 10% of the data for testing purposes. To compensate for the lack of data, we implement several pre-trained architectures and fine-tune their abstractions over our dataset to find the most effective solution.

For the colorizing model, we explore usage of U-net architectures starting from a basic U-net model and experiment with different color spaces and empirically confirm the superiority of L*a*b color space in image colorization problem. We use the state of the art ResNet-18 to provide as a backbone of the encoder and build a U-net around it. The pre-trained network over COCO dataset in RGB colorspace produces significantly weaker results as compared to the subsequent network in L*a*b colorspace. The best performing model turns out to be the ResNet18 U-net which is fine-tuned over our particular dataset to produce appealing and similar results to the ground truth.

The Super-resolution model is based largely on the SR-GAN proposed by Ledig et al. (2017). We use the generator weights and sample results from the training set to inspect the results. It is found that the model performs really well on the pre-trained weights and we decide to fine-tune it to our application. After fine-tuning the model, we train other state of the art single image super-resolution models such as EDSR (Lim et al.; 2017) and WDSR (Yu et al.; 2018). These provide further insights into the problem and simultaneously, improve the results.

While studying and improving the performance of these models, we explore performance metrics of GANs and evaluation methodologies implemented to test out conditional GANs. It is evident that the loss curves of generator and discriminator do not provide us with any intuition about the model performance. We also discover that standard distance metrics cannot be used to evaluate GANs and quantitative methods that exist to evaluate GANs are unreliable. We prove so by contradiction of qualitative samples and quantitative measurements of the best performing architecture for colorization models. However, we observe that quantitative estimation is quite reliable for the problem of single image super-resolution and can be helped to determine which model is better suited for the task.

VI. Future Scope

Though we obtain moderately good results, a vast amount of algorithms still remain unscratched. A more powerful model such as SE-ResNext, EfficientNet and more such state-of-the-art models can be implemented and trained over millions of images from the Imagenet. With even more hardware resources and availability of data, we can explore computationally heavy models for a better approximation. An image stitching algorithm can be applied on the produced images to generate large scale astronomical images for scientific study. Colorization can be improved by the virtue of exploring different loss functions using weighted losses to reduce loss problem for low saturation regions. We can introduce a gradient penalty for the SRGAN architecture and include the WGAN (Arjovsky et al.; 2017) which will stabilize the discriminator to stay within the space of 1-Lipschitz function. Progressively growing GANs (Karras et al.; 2018) can be applied so that the dimensions can be further improved with more stability and greater sharpness.

References

Arjovsky, M., Chintala, S. and Bottou, L. (2017). Wasserstein gan.

Cheng, Z., Yang, Q. and Sheng, B. (2016). Deep colorization.

Dahl, R. (2016). Automatic colorization.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial networks.

Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J. and Murphy, K. (2017). Pixcolor: Pixel recursive colorization.

He, K., Zhang, X., Ren, S. and Sun, J. (2015). Deep residual learning for image recognition.

Huang, Y.-C., Tung, Y.-S., Chen, J.-C., Wang, S.-W. and Wu, J.-L. (2005). An adaptive edge detection based colorization algorithm and its applications, pp. 351–354.

Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A. A. (2018). Image-to-image translation with conditional adversarial networks.

Karras, T., Aila, T., Laine, S. and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. and Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network.

Levin, A., Lischinski, D. and Weiss, Y. (2004). Colorization using optimization, *ACM SIGGRAPH 2004 Papers*, ACM Journals, pp. 689–694.

Lim, B., Son, S., Kim, H., Nah, S. and Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution.

Long, J., Shelhamer, E. and Darrell, T. (2015). Fully convolutional networks for semantic segmentation, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Peh, G. X. and Marshland, K. (2019). Astronomical image colorization and super-resolution using residual encoder networks and gans.

Qu, Y., Wong, T.-T. and Heng, P.-A. (2006). Manga colorization, *ACM Transactions on Graphics (TOG)* **25**(3): 1214–1220.

Radford, A., Metz, L. and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Tola, E., Lepetit, V. and Fua, P. (2008). A fast local descriptor for dense matching, *Proc. CVPR* .

Tom and Katsaggelos (1996). Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images, *in* Anon (ed.), *IEEE International Conference on Image Processing*, Vol. 2, IEEE, pp. 539–542. Proceedings of the 1995 IEEE International Conference on Image Processing. Part 3 (of 3) ; Conference date: 23-10-1995 Through 26-10-1995.

TSAI, R. (1984). Multiframe image restoration and registration, *Advance Computer Visual and Image Processing* **1**: 317–339.
URL: *https://ci.nii.ac.jp/naid/10026807118/en/*

Yatziv, L. and Sapiro, G. (2006). Fast image and video colorization using chrominance blending, *IEEE Transactions on Image Processing* **15**(5): 1120–1129.

Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X. and Huang, T. (2018). Wide activation for efficient and accurate image super-resolution.

Zhu, J.-Y., Krähenbühl, P., Shechtman, E. and Efros, A. A. (2018). Generative visual manipulation on the natural image manifold.