Colourisation of GrayScale Images for Wildlife Tracking

Aanchal Narendran
PES1UG19CS006

A Narendiran
PES1UG19CS001

Abdul Rahman
PES1UG19CS009

Abhishek D
PES1UG19CS020

***Abstract*—The project aims to colourise grayscale images using Image Processing, Machine Learning and Deep Learning techniques. Colourising an image is one such problem that doesn't have a clear cut solution or approach. A wide range of authors have developed novel Convolution based approaches, Auto-Encoder based models and so on for the same. This project aims to enhance images using Filtering, Feature Extraction, Colour Space mapping to effectively model the data being passed to a Deep Learning Model to colourise the image.**

***Index Terms*—Image Processing, Deep Neural Networks, Image Colourisation, AutoEncoder**

I. Introduction

Image colourisation involves using a wide range of statistical models to estimate the red, green and blue values of pixels in a grayscale image. There is an ever-present research intrigue in the field of colurisation owing to it's open-ended nature. Most photographic solutions between 1826 - 1980 have relied on capturing their images in Grayscale. Colourising these images is similar to art restoration. Although old, they are of importance and hold sentimental, historic value and scientific value. CCTV solutions over the past decades are usually recorded in grayscale images. Effectively limiting the usage of these cameras in identifying vehicles and people of interest in legal cases. Although luminance and grayscale help isolate information regarding the objects in the image, colour helps improve the quality of information obtained from grayscale images. This project aims to develop a solution to aid in colourising images in one such domain.

Due to the complexity of colourising images, there is no standard solution for this problem. This project explores a wide range of models and image processing methods to enhance estimates of colour and improve on present colourising solutions. The present solution is to exploit the Animals 151 dataset and to explore and validate colourisation strategies on this data where the colourised version serves as a benchmark. Wildlife surveying is of key interest to track evolution and monitor species count. Especially in the cases of near-extinct to in-danger species, wildlife reserves constantly keep track of the count of the species. Due to the length and breadth of these images, often greyscale cameras are used over their colourised counterparts. The current solution will not only assist in colurising these greyscale image but would also help in improving those images caught in the night or in low forest ground of dense forests with low illumination.

II. Literature Review

A. Image Colorization Using Similar Images

The paper [2] leverages exemplar images to map the RGB values using superpixels. Exemplar images are coloured images that are semantically similar to the target image (the exemplar image has elements present in the target image). The author uses superpixels to map a region from the exemplar image to the target (grayscale) image. The paper addresses a multiple exemplar image model to colourise any grayscale image. Superpixel extraction is done to modify the shape and size of the image while retaining the original edges. SURF and Gabor feature easily discriminate between superpixels. The intensity and the standard deviation is easily computable and available but they give 4 parameters for discrimination. Whereas, SURF and Gabor give the model 128 and 40 features respectively to enhance discrimination. Fast cascade feature matching along with euclidean distance is used to compute the similarity between a superpixel from the exemplar image and the target image. A voting strategy in clustering was implemented to help fix incorrect colour mapping. K-Means clustering was used to differentiate between colour mapping. Dense clusters have near-accurate colour mapping. The images obtained were evaluated by a panel of volunteers to judge correctness.

The authors were able to back their superpixel assumption with sound results where 65% of a panel of 30 volunteers categorised their image as real, as opposed to the nearest benchmark at 53.2%. Due to superpixelation, the model struggles with handling areas with a sharp increase in intensity or around the edges. It leads to bleeding artefacts. The model is heavily reliant on the availability of exemplar images. However, the model is nested heavily in standard morphological processes making it easier to fix an issue in the image as opposed to black-box models.

B. End-to-End Conditional GAN-based Architectures for Image Colourisation

The existing convolutional neural network impacts the colourfulness of an image. This paper aims uses Generative Adversarial Networks to strengthen colour mapping.

Additionally, integration of instance and batch normalisation further help generalise. The Generative Adversarial Networks trained on imageNet model the colour distribution and to make the samples appear computationally to natural images. Spectral Normalisation methods further improve generalisation without hampering the stability of training by regularising the weights. Multi-scale discriminators improve generation in small areas and enhance local details. CIELab space is used as it is more perceptually uniform and linear. The generator and discriminator are trained using min-max to obtain nash equilibrium. The generator captures the original distribution of data while training a realistic mapping. The generator is modelled towards generating images that are harder for the discriminator to differentiate. The U-Net is used as a generator while Markovian PatchGANs are used as discriminators.

The author of the paper uses a pure Black-box model to colourise the image. This doesn't leave a lot of space for handling noise and other image artefacts. Noise in datasets has a huge role in successful feature extraction and segmentation. However, the author is successful in demonstrating that Generative Adversarial Networks perform better in terms of colourfulness.

C. Automatic Colorization of Grayscale Aerial Imagery Using XGBoost Regression

This paper [4] aims to use automatic colourisation to isolate those areas which haven't had any changes over time by using Pseudo-Invariant Features. Like wildlife survey images, most aerial images used for city mapping and town planning in the past were generated in the greyscale format. The authors of the paper plan on using colourised versions of the greyscale images to help understand the development of the city. The paper also highlights how the prediction of direct R, G, and B values is a much more complex task than colourising images using greyscale values. The author mentions the usage of statistical measures such as median, mean, maximum and so on to account for the impact of the neighbourhood of a pixel on the pixel in question. The results of the paper are evaluated by using the Peak Signal-to-Noise Ratio. It is used to evaluate image quality post-processing of an image.

Although the authors of the paper perform colourisation of grayscale images, the main goal of the paper is to map unchanged regions using pseudo-invariant features. Although the authors produce a visually similar image, the noise from the original image gets reflected in the produced image impacting the PSNR metric.

III. Proposed Solution

A. Dataset

The dataset "Animals 151" will be used in this project. It contains RGB pictures of about 151 animals, with each animal having anywhere between 30 to 60 images. Each image is of the size 224x224 pixels **??**. One of the working assumptions of this project is that these colour images will be able to effectively model the actual scenario. Most animals are found in the wild and are often captured from varied backgrounds. We hope that with effective feature detection and modelling the solution effectively catches any animals of interest in the frame. For the sake of this project, subject to compute restraints the dataset size has been constricted to 60 animals instead of the original 151.

B. Preprocessing

As a part of preprocessing, our initial step was to segment the dataset to ensure the presence of enough samples to effectively model the problem statement. We decided to use data obtained from 60 animals as opposed to the 151 that composed the dataset. This decision to use a subset of the data is driven by both computational and storage efficiency. As a part of preprocessing, the first step was to build the dataset of RGB-coloured images. However, the RGB colour space was not directly worked on owing to the added complexity of predicting RGB values. The CIE-LAB colour space, however, was chosen. This decision is because, in the LAB colour space, the L component is almost synonymous with the intensity values of a greyscale image. This fact is used to the advantage to model the a and b values to give the LAB space of the coloured version of the input greyscale image. Apart from this, preprocessing was limited since both Convolutional Neural Networks and AutoEncoders are models that perform better feature engineering than human trial-and-error. However, the data had to be flattened and reshaped before passing them to the model.

In the case of the XGBoost Regressor, a certain amount of preprocessing is required. Unlike the aforementioned models, the XGBoost Regressor does not perform feature engineering. A Colour space of 128 is defined to which the pixels of the training colour images are mapped. The reasoning behind this is that the original coloured images are capable of representing over 11000 different colours but modelling this would be too compute expensively and near impossible without additional compute resources. As a part of training, the A and B components of the LAB space of the inputted colour images are obtained and passed to a quantiser developed using KMeans. Each (A, B) pair is mapped to a centroid during K-Means clustering which in return is mapped to a colour space in the 128-colour space. During testing, it is the A and B values and the colour map of the pixels that are predicted.

C. Models

As a part of Modelling phase of the project, three models were developed to compare against each other. The three models in question are a XGBoost Regressor, a Convolutional Neural Network and an AutoEncoder. Over the course of this section, these models and their development is looked into at depth.

1) XGBoost Regression: XGBoost Regressors are commonly used in a wide variety of Computational Modelling owing to their superiority in effectively modelling Deep Learning and Machine Learning problems. In addition, the paper [4] uses XGBoost Regression to colourise aerial images. The authors of the paper can successfully colourise the dataset of aerial survey images to an extent. Moreover, Aerial Survey images are about as diverse as the coloured versions of images of wildlife. They also account for a wide variety of backgrounds around the objects of interest. With this in mind, the XGBoost regressor was the first model implemented as a part of the problem statement.

In the previous subsection, an in-depth analysis was provided of the preprocessing of the XGBoost Regressor. This level of pre-processing is crucial since the development of the model on its own is not an overly complicated process. The existing XGBoost package was used in the development of the model. Two XGBoost regressors are built however only one is used. As a part of Regression, the first step is to build the dataset with the L, A, and B values along with the features of importance. The quantised (A, B) values produce a colour map. This colour map is treated as a label. The training data to the Regressor are the extracted featured using Histogram of Gradients and Daisy feature descriptor based on SIFT. The extracted features along with labels from the colour map. Post-training, one grayscale image at a time is passed to obtain predictions and the comparison as mentioned in the evaluation.

2) Convolutional Neural Networks: Convolutional Neural Networks are the standard for modelling any Image Classification or Processing problem. Keeping in line, the second phase of the project involved modelling a convolutional neural network to solve the problem. Artificial Neural Networks weren't explored before this owing to the high number of trainable parameters and heavy computations. The Activation function combined with convolutional filters and striding effectively helps extract a feature map of much higher importance than human trial and error. Like the XGBoost Regressor model, the input training data to the convolutional neural network model was the coloured image in LAB colour space.

The convolutional neural network for the project was developed entirely using the PyTorch model of Python. For the model, the standard ResNet18 architecture with modifications was used. Due to the complicated nature of the problem statements and existing solutions using very deep neural networks, the ResNet18 architecture was chosen since it used residual blocks to its benefit without increasing the overall depth and the number of trainable parameters of the neural network. The first half of the model followed the ResNet architecture. The output of the ResNet architecture was passed to another deep neural network with Convolution, RELU and BatchNormalisation layers. During convolution, striding and passing were performed with the Relu activation function. Batch Normalisation was performed to handle the overfitting of the model. The model was trained for over 300 epochs.

3) Auto-Encoder: AutoEncoders are popularly used in the Image Processing domain of Machine Learning. Their popularity stems from the fact that images are huge chunks of data which require some level of feature selection. Performing Principal Component Analysis or Correlation Analysis to choose features of importance which are usually done in regression problem statements is not applicable in this case since PCA and Correlation would not account for the spatial importance of the pixels being dropped. AutoEncoders work on the concept of Encoding and Decoding data to build a representation that can be trained on and whose results can be used to develop predictions.

As a part of the autoencoder implementation in the problem, a level of reshaping is performed. Post-Reshaping, the L, A and B components of the coloured images are extracted and combined into a joined dataset to pass to the encoder. As a part of encoding, the image is passed through numerous convolutional layers to extract the embedded version of the images. The convolutional filters get bigger as the images passes down the network. The decoding layer has successive convolutional layers followed by an upsampling layer to maintain dimensionality with relu activation.

IV. Experimental Results

The experimental results of the three models are detailed in this section. In a wide majority of colourisation application, a panel of volunteers are used to distinguish between the actual coloured versions of the picture and generated versions of the picture. Due to the lack of volunteers, this paper relies purely on the authors' visual perception and quantitative metrics to judge the results of the models.

To quantitatively judge the model, we use Histogram similarity comparison. To do this, the original image and the predicted image are flattened and histograms of the same are obtained. The histogram comparison module is used on three images to understand the performance of the model. The histograms are evaluated based on the Chi-Square distance score and Intersection Metric. The smaller the Chi-Square distance metric the more the histograms match indicating that the predicted picture is closer to the original coloured picture. In tandem, the Intersection score is used to compare how much the histograms match each other. The larger the intersection score the more similar are the images in terms of colour. In the table I, the results from the XGBoost Regressor are documented where the Chi-Square value is high due to the limitation imposed on the colour feature space. In the table II, the results of the convolutional neural network are documented. In the table III, the results of the AutoEncoder are documented. From a purely quantitative perspective, it is observed that the AutoEncoder performs significantly better than the Convolutional Neural Network and the XGB Regressor. The lack of performance of the XGBRegressor could also be related to the lack of neighbourhood information being included in the training data which is adequately captured by the convolutional feature maps in the Convolutional neural network and AutoEncoder.

Now, the models are assessed on a qualitative basis based on the images obtained from the models' post-training. First, we look at the images obtained from the XGBoost Regressor. In the figure 1, we see that the black and white parts of the original image are modelled properly however there is a

TABLE I
XGBOOST REGRESSOR

Image	Chi-Square	InterSect
Ailurpoda-melanoleuca	11037	0.308
Ceratotherium-Simum	7548	1.67
Dasypus-Novemcinctus	4.8	0.36

TABLE II
CONVOLUTIONAL NEURAL NETWORK

Image	Chi-Square	InterSect
Ailurpoda-melanoleuca	60.4	1.5
Ceratotherium-Simum	40.03	0.9
Dasypus-Novemcinctus	53.06	1.59

TABLE III
AUTO ENCODER

Image	Chi-Square	InterSect
Ailurpoda-melanoleuca	2.04	1.8
Ceratotherium-Simum	0.6	2.6
Dasypus-Novemcinctus	2.4	0.38

slight blue tinge in the image and there is the presence of contours. The occurrence of contours is due to the decrease in the number of pixels being used to model the colour space due to computational constraints.

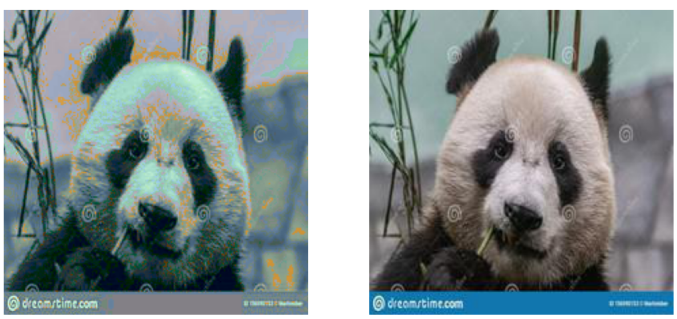

Fig. 1. Ailurpoda-melanoleuca predicted by the XGBoost Regressor

Fig. 2. Colourisation Ailurpoda-melanoleuca by the Convolutional Neural Network

In the image 2, the generated image has the same blue tinge problem faced by the XGB Regressor but the image is devoid of the contouring caused in the XGB Regressor. To understand the problem, a review of the images was conducted and observed that the cause of the bluish tint in both images is due to an issue in the White Balance of the image causing the image to look more "Blue" than expected. Research into understanding how to fix white balance leads to the conclusion that white balance needs to be fixed while taking an image of the object. A few quantisation and saturation strategies were experimented with to fix the bluish tint, however, the efforts did not yield significant improvements.

Fig. 3. Colourisation Ailurpoda-melanoleuca by the AutoEncoder

In the image 3, the generated image is much closer to the original image of the panda with the image being much smoother than all the previous counterparts. However, we notice that contrary to the initial images, the AutoEncoder image has a yellow tint problem. The original survey into the issue with the images helped conclude that this was also caused due to the white balance problem in the photographs. A few corrective measures to saturation were experimented with to fix the results of the image.

Through the experiments of the paper, a founding step in the strategy to model wildlife animals in survey images has been approached. Despite colour fluctuations, the models can represent the structural similarity of the original images to a tune of 85 to 90% accuracy. Further advanced understanding of the Colour Theory might be warranted to fix the white balance issues faced by the images. There exists ready-made software in photoshop to modify the white balance to a helpful extent. However, replications of the software weren't feasible. Furthermore, an improved dataset with higher resolution images might aid in modelling this problem better. The lack of labelled images about wildlife surveys online resulted in using data to simply colourise animals.

V. CONCLUSION

In conclusion, the paper shows that the above models could be used to an extent to colourise wildlife and images to a certain accuracy. Out of the three models developed, AutoEncoder networks proved to be most successful in colourising the image with a slight yellow tinge issue. The issue in the output images was caused due to White Balance which the authors of the project weren't equipped to solve. However, further research into the use case and methodology is fully warranted to develop a solution which accounts for and fixes the white balance issues.

VI. ACKNOWLEDGMENTS

We place on record our gratitude to Dr. Gowri Srinivasa and the entire Image Processing and Computer Vision course team, for motivating us and providing us with this opportunity to get hands-on experience in the domain. This project has proven

to be a great experience in learning how work with images and understanding truly how complex something as simple as a picture can be. We hope to work on the project in the future to further our findings and establish them fully.

References

[1] R. Pucci, C. Micheloni, N. Martinel, "Collaboration among Image and Object Level Features for Image Colourisation.", ArXiv, 2021

[2] R. K. Gupta, A. Y. Chia, D. Rajan, E. S. Ng, H. Zhiyong, "Image Colorization Using Similar Images", Proceedings of the 20th ACM international conference on Multimedia, 2012

[3] M. G. Blanch, M. Mrak, A. F. Smeaton, N. E. O'Connor, "End-to-End Conditional GAN-based Architectures for Image Colourisation", arXiv, 2019

[4] H. J. Kim, Y. D. Eo, D. K. Seo, J. S. Park, G. W. Park, W. Y. Park, "Automatic Colorization of Grayscale Aerial Imagery Using XGBoost Regression", Proceedings of 40th Asian Conference on Remote Sensing, 2019

VII. Appendix

A. Contributions

1) Aanchal Narendran: Literature Survey, Preprocessing, XGBoost Regression with K-Means quantisation
2) A Narendiran: Convolutional Neural Network
3) Abhishek Dinesh: Convolutional Neural Network
4) Abdul Rahman: AutoEncoder

B. Additional Visualisation

- In 4, we see Ceratotherium-Simum colourised using the XGBoost Regressor
- In 5, we see Ceratotherium-Simum colourised using the Convolutional Neural Network
- In 6, we see Ceratotherium-Simum colourised using the Auto Encoder
- In 7, We see Dasypus-Novemcinctus colourised using the XGBoost Regressor
- In 8, We see Dasypus-Novemcinctus colourised using the Convolutional Neural Network
- In 9, We see Dasypus-Novemcinctus colourised using the AutoEncoder

Fig. 4. Colourisation Ceratotherium-Simum by the XGB Regressor

Fig. 5. Colourisation Ceratotherium-Simum by the CNN

Fig. 6. Colourisation Ceratotherium-Simum by the AutoEncoder

Fig. 7. Colourisation Dasypus-Novemcinctus by the XGB Regressor

Fig. 8. Colourisation Dasypus-Novemcinctus by the CNN

Fig. 9. Colourisation Dasypus-Novemcinctus by the AutoEncoder