# The Greatest Teacher, Failure Is: Evolving a Genetic Program to Colour Greyscale Images


Liam McDevitt
6015929
Computer Science
Brock University
St. Catharines, Canada
lm15ue@brocku.ca

Andrew Pozzuoli
6017735
Computer Science
Brock University
St. Catharines, Canada
ap15yl@brocku.ca



## I. Abstract

*This paper looks at applying genetic programming (GP) to colour greyscale images. Three different fitnesses were explored. One was using the hits and sum of errors to match each pixel's hue similarly to a regression. This often led to GP picking the most popular colour of the image and applying it as a filter. The second was to only take into account the colour distribution of the entire image and try to generate a similar amount of each colour. This tended to create noisy images of randomly placed colours. The third approach was combining the first two to try to keep similar colours together which resulted in more uniform, yet slightly noisy and interesting looking colourings. For each of these fitnesses, two training sets were used to see if that affects the colouring of the testing sets. Two testing sets were used, one was a similiarly coloured image to the training set and one was differently coloured. It was found that training set affects the colourings of the testing set where colourings were often based on the training set colours.*


## II. Introduction

The goal of this set of experiments was to use genetic programming (GP) to evolve a function that uses a given set data relating to the pixel of a greyscale image and its surrounding pixels and generates a hue for that pixel. The input greyscale image is generated by taking the original coloured image and separating it into three channels: hue, saturation, and brightness. The brightness by itself gives a greyscale image and the GP is to output the value for the hue channel.

One of the challenges of colouring a greyscale image is that there is no information in the image itself that could hint at the actual colouring. If a human were to attempt to recolour a greyscale image, the colour choices would need to be informed by experience seeing in colour. In other words, given a black and white image of a carrot, a human would colour it orange because they would have seen a carrot before. Otherwise, there is no information regarding what colour the carrot would be. From this, the colouring has an element of subjectivity that can be difficult to measure with a fitness function which is why the different fitness functions were tested in the experiments.

The fitnesses tested were inspired by the fitnesses used in Machado, Dias, and Cardoso's paper, *Learning to Colour Greyscale Images* [4].

## III. Experimental Setup

### A. Data

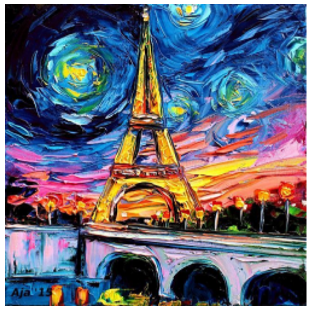
Two training sets were used to test how the training image affects the colouring of similarly and differently coloured images. Each training set came from the pixel data of an image. The data points described the intensity of every pixel as well as some information in an 11x11 area filter surrounding the pixel. Each testing image was 250x250 pixels resulting in a total of 62500 pixels. There were also two testing sets, each a 250x250 image. One of the testing sets was similarly coloured to one of the training sets and the other testing set was similarly coloured to the other training set. This way we can observe how the GP generalizes onto an image with a similar colouring as well as one with a different colouring.

The intensity was calculated by taking the average of the RGB colour values of every pixel. This gave a number between 0 and 255. The hue channel of every pixel was a number between 0 and 1 representing a position on a colour wheel (a percentage of 360°).

The training data came from two images. One was a still frame from the Disney show: The Mandalorian [6]. This was used to represent a data set with a low colour distribution. The other image was a vibrantly coloured painting with a higher distribution of colours [8]. The two testing images were similar. One was a still from Star Wars V: The Empire Strikes Back [1] with a low colour distribution and the other image was a similarly coloured painting from the same artist [**?**]. The hypothesis is that training and testing done on similar images would yield better results than training on a similar image and testing on a differently coloured image.

In order to recolour the image with the output hue value, the image also needs the brightness and saturation of each

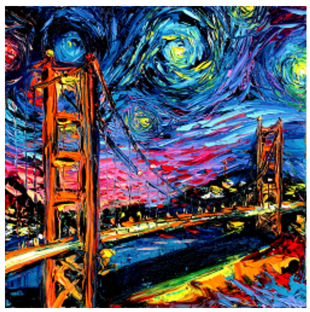
(a) Training Image 1: A still frame from Disney's The Mandalorian (Baby Yoda)



(b) Training Image 2: Painting by Aja Kusick (Painting 1)

Fig. 1: Training Images

(a) Testing Image 1: A still frame from Star Wars V: Empire Strikes Back (Yoda)



(b) Testing Image 2: Painting by Aja Kusick (Painting 2)

Fig. 2: Testing Images

pixel. The brightness channel and saturation channel had to be stored separately in order to be used when recolouring the image with the output hue. The pixels of greyscale images contain information on brightness but nothing on saturation since saturation is a property of colour. The consequences of this will be discussed in the conclusion.

### B. *GP Language*

The data sets for training and testing were precomputed to speed up runtime. The GP did not calculate the pixel brightness nor the area filters. These values were put into a text file which the GP read line by line to create terminals.

### C. *Parameters*

The maximum tree depth was restricted to 7 and the maximum grow depth was restricted to 4 to improve run time. The population was also restricted to 100 to improve run time. The crossover rate was 70% and the mutation rate was 25%. This was determined during preliminary runs to try and get more interesting colourings. Higher mutation allowed for more exploration in an attempt to get away from simply selecting safe options that filter the image based on the most popular colour.

TABLE I: Terminals

| **Terminal** | **Description** |
|---|---|
| Intensity | The value of the intensity of the pixel |
| Max Intensity 11x11 | The maximum intensity value for a pixel in the 11x11 area filter surrounding the pixel |
| Min Intensity 11x11 | The minimum intensity value for a pixel in the 11x11 area filter |
| Mean Intensity 11x11 | Average intensity of all pixels in the 11x11 area filter |
| Standard Deviation Intensity11x11 | The standard deviation of the intensity of all pixels in the 11x11 area filter |

TABLE II: GP Paramters

| **Parameter** | **Value** |
|---|---:|
| Population Size | 100 |
| Generations | 50 |
| Population Initialization | Ramped Half and Half |
| Minimum Grow Depth | 2 |
| Maximum Grow Depth | 4 |
| Maximum Tree Depth | 7 |
| Selection | Tournament, k=4 |
| Crossover | Subtree Crossover, 70% |
| Mutation | 25% |
| Data Points Training | 62500 |
| Data Points Testing Set 1 | 62500 |
| Data Points Testing Set 2 | 62500 |
| Runs Per Experiment | 10 |

### D. *Functions*

The functions used were similar to those of the paper by Machado et al. with the exception of a few. The simple arithmetic operators (addition, subtraction, multiplication, and division) were used to keep complexity low. Negative, square, and square root were added and sin, cos, if, and get were removed. If and get were not implemented to keep complexity low and reduce risks of bloat. Sin and cos were tested in preliminary runs but were removed after it was discovered that these functions were often abused by the GP since they can easily be modified to get a periodic function that stays between 0 and 1. These periodic functions would yield very high fitnesses but would be filled with noise they were thus removed during experimentation in an attempt to yield more uniform colourings.

### E. *Fitness Evaluation*

Three methods for fitness evaluation were tested. Pixel hue regression, colour distribution, and hybrid.

*1) Pixel Hue Regression:* This took the GP output hue and compared it to the actual pixel's hue from the original coloured image, similar to a regression. To calculate this, the hit count was incremented by one if the GP output hue was within 5% of 360°or 0.014 of the actual hue. The sum of hits increased by one if the following was true:

$$|actualHue_i - GPOutputHue_i| <= 0.014$$

TABLE III: Functions

| Function | Description |
|---|---|
| Add | Adds two terminals |
| Multiply | Multiplies two terminals |
| Subtract | Subtracts two terminals |
| Divide | Divides two terminals<br>If the denominator is zero then<br>set the denominator to 1 |
| Negative | Makes the terminal negative |
| Square | Multiplies the terminal value with itself |
| Square Root | Takes the square root of a terminal<br>If the terminal is negative then<br>take the square root of<br>the absolute value of the terminal |

If this was found to be true, the error for that pixel was set to 0 otherwise the error is set to that absolute difference $>$ 0.014. The higher number of hits results in a better fitness.

A sum of errors was also calculated as follows:

$$sumOfErrors = \sum_{p \epsilon P} p_{error}$$

Where $p_{error}$ is the absolute difference in the actual hue and the GP output hue according to the above error calculation. This gave the standardized fitness where a value of 0 is ideal and higher values are worse.

This fitness favours getting the right colour in the right location.

*2) Colour Distribution:* This fitness took original coloured image and put every pixel into 16 buckets depending on where on the colour wheel it lies. A hue that is less than $\frac{1}{16}$th goes into the first bucket, if it is greater than $\frac{1}{16}$th and less than $\frac{2}{16}$ths, then it goes into the second bucket and so on. This gives an array of colour distribution with each bucket showing the number of pixels in a spectrum of similarly coloured pixels. A similar array was created for the output the GP gave where every pixel is put into a bucket based on its location on the colour wheel. Once all pixels have been placed in the buckets, every bucket is compared and the hits are incremented according to the following formula:

$$hits = \sum_{i=0}^{15} (1 - error_i)$$

Where $error_i$ is the normalized error of each bucket in the GP colour distribution according to the following formula:

$$error_i = \frac{|truthColourSpread_i - gpColourSpread_i|}{truthColourSpread_i}$$

Where $truthColourSpread_i$ is the number of pixels in the ith bucket of the actual colour distribution array and $gpColourSpread_i$ is the number of pixels in the ith bucket of the GP output colour distribution. To avoid divide by zero, if $truthColourSpread_i$ = 0 and $gpColourSpread_i > 0$ then $error_i$ is set to 1. If $error_i > 1$ then it is set to 1. This way, hits is never negative.

This fitness will reward if the same number of pixels in each bucket is the same between the actual image and the GP output colouring. Since it is normalized, even 5 pixels in one bucket has the same weight as 1000 pixels in another bucket to avoid dominance. This way the colour distribution is rewarded if they are similar. Note that this fitness does not take into account the location of the pixels at all. Due to normalization, the highest number of hits can only ever be 16 since each bucket's hit is 1 - error. Thus less error gets closer to 1. During statistics calculations, the final hit count is cast to an integer which removes anything after the decimal.

The standardized fitness is simply the sum of the error values.

*3) Hybrid:* This fitness uses both of the previously described fitnesses working together. The hit count for the regression version can be considerably higher due to the number of pixels so to avoid the regression dominating the distribution, the hit count from the regression fitness was normalized. After the hit count from the regression was calculated, the hit count was divided by the input data set size to get the average amount of correct hits. This value was then multiplied by 16 to hold the same amount of weight as the distribution fitness. In other words, the maximum hit count after this regression adjustment is 16 which is the same as the maximum hit count after just the distribution. Distribution was calculated the exact same way as described before and their hits were added together to get a total hit count out of 32. Sum of errors was calculated the same way with an adjustment after the regression calculation which took the sum of errors, divided by the set size, and multiplied by 16.

This fitness rewards similar colour distribution as well as getting the right location.

The adjusted fitness took the percentage of hits with each function and subtracted this from 1. This gives a number between 0 and 1 with 1 being the worst and 0 being ideal.

## IV. Experiment

The experiments analyzed how different fitness evaluations and training sets affected the GP's ability to colour a greyscale image.

For fitness, the three fitnesses described in the previous section were tested across ten runs.

Within each of these fitness experiments two further experiments were conducted. One where the training set was Baby Yoda (figure 1. a) and one where the training set was Painting 1 (figure 1. b). The best individual from both training sets were tested on both testing images (figure 2) to see how training affects colouring similar and different images.

The GP output was combined with the brightness and saturation of the testing images to colour the image. In addition to fitness as evaluation, these colourings were evaluated subjectively with a "thumbs up, thumbs down" approach of evaluation according to the researchers' opinions of what a "good" colouring is. This way, the fitness could be evaluated against each other. Otherwise, since each fitness gives a different method of evaluation, comparing across fitnesses with a purely mathematical approach is difficult to do. Fitness only gives the best according to its own function.

## V. Statistical Analysis

Unpaired t-tests with unequal variance were performed on different sample means from the best fitness values of each generation for our three different fitness functions: pixel hue, colour distribution, and a hybrid. We are unable to compare the performance over the three different functions statistically due to their output not being standardized relative to one another. Instead we are checking for statistical significance within each fitness function between training on an image with a low colour distribution and an image that has many different colours within it. The sample means are based on the average of the population's best fitness values over our 10 runs. We will say that for all t-tests conducted the null hypothesis is that the sample means are equal and the alternate hypothesis is that they're not equal.

LCD = Low colour distribution
HCD = High colour distribution

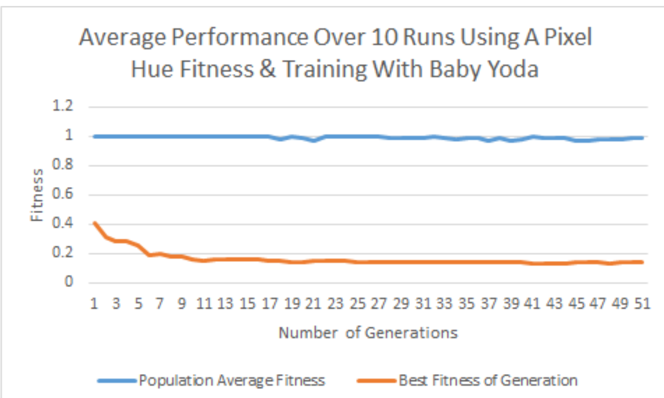



Fig. 3: Pixel hue regression average fitness and best fitness averaged over ten runs trained on Baby Yoda

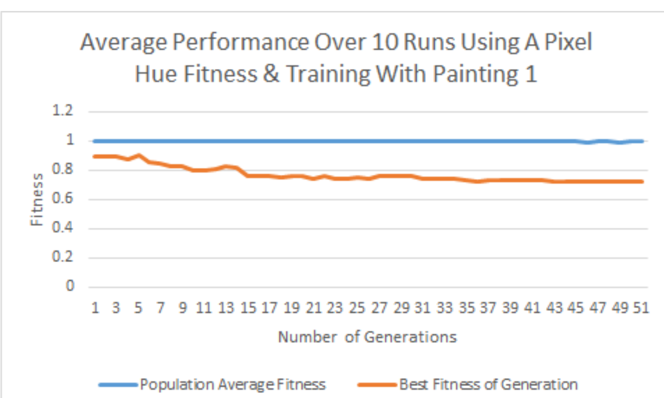



Fig. 4: Pixel hue regression average fitness and best fitness averaged over ten runs trained on Painting 1

### A. Pixel Hue: LCD Training vs. HCD Training

There is very strong evidence ($p = 1.223\,789 \times 10^{-78}$) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled mean comparing an image which has a low colour difference between its pixels and an image which has a high colour difference between its pixels for our pixel hue fitness function.

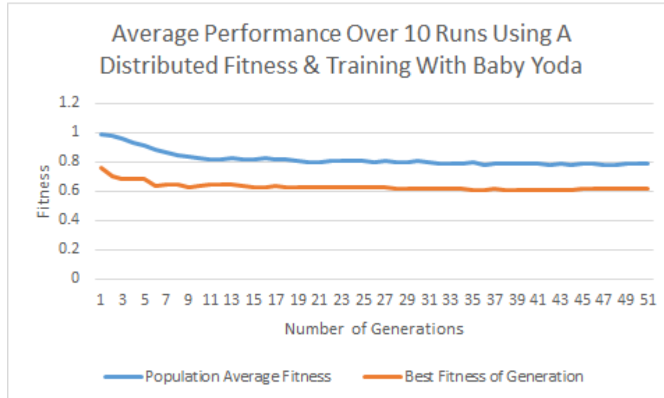



Fig. 5: Colour distribution average fitness and best fitness averaged over ten runs trained on Baby Yoda

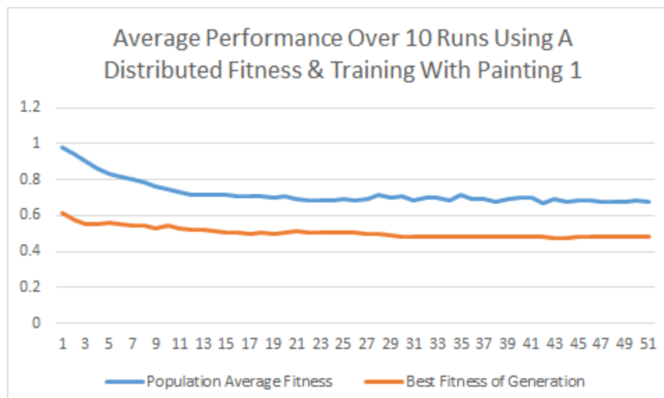



Fig. 6: Colour distribution average fitness and best fitness averaged over ten runs trained on Painting 1

### B. Colour Distribution: LCD Training vs. HCD Training

There is very strong evidence ($p = 3.053\,695 \times 10^{-39}$) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled mean comparing an image which has a low colour difference between its pixels and an image which has a high colour difference between its pixels for our colour distribution fitness function.

### C. Hybrid: LCD Training vs. HCD Training

There is very strong evidence ($p = 3.894\,575 \times 10^{-10}$) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled mean comparing an image which has a low colour difference between its pixels and an image which has a high colour difference between its pixels for our hybrid fitness function.

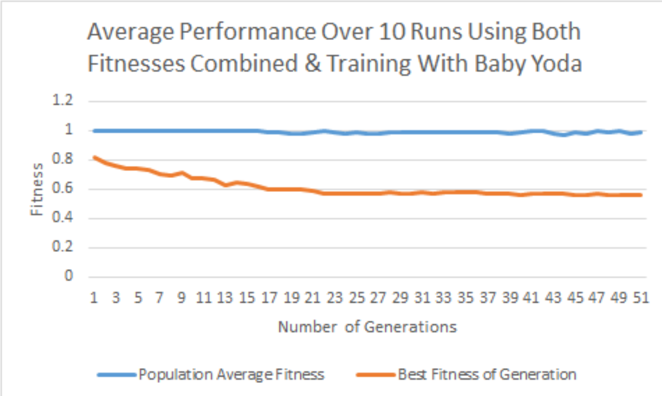



Fig. 7: Hybrid average fitness and best fitness averaged over ten runs trained on Baby Yoda

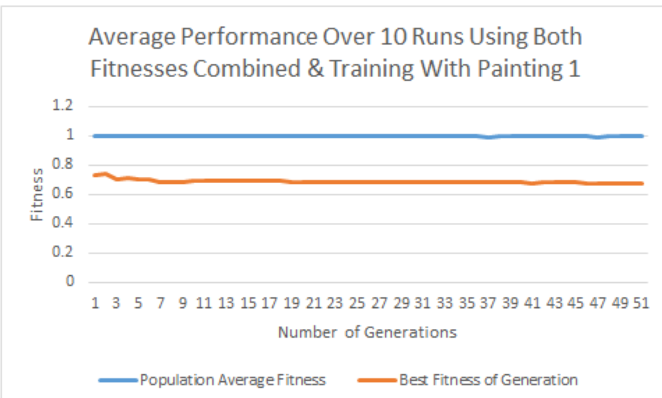



Fig. 8: Hybrid average fitness and best fitness averaged over ten runs trained on Painting 1

## VI. RESULTS

The actual colouring of the images and the subjective nature of what a "good" colouring is does not lend itself well to statistical analysis. This section will display how the fitness and training set experiments coloured the testing images.

All best trees used to colour these images can be found in the appendix.

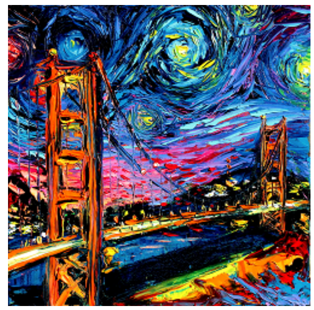
With the pixel hue regression fitness, it seems that whatever the most popular pixel hue was from the training set, that pixel hue is applied as a sort of filter on the testing image. It also seems that with this type of colouring, the testing images with a lower colour distribution will appear subjectively better coloured than the higher distribution of the painting images.

The distribution fitness seems to create subjectively poorly coloured images. There is a tendency to put random splashes of colour in weird spots with little pattern. It is possible that all of the colours are present in the image contributing a good fitness however it is not pleasing to the eye.

The hybrid fitness seems to colour the image with less noise than the colour distribution fitness but still noisier than the regression. It colours the image more uniformly than the colour distribution but with more variety in the colour choice

(a) Testing Image 1: Ground Truth

(b) Testing Image 1: GP Colouring

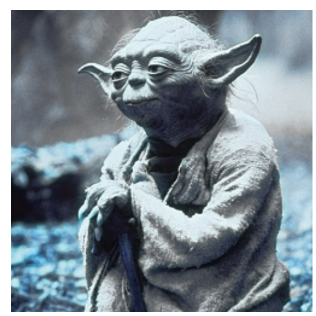


(c) Testing Image 2: Ground Truth

(d) Testing Image 2: GP Colouring

Fig. 9: GP colouring with pixel hue regression fitness and trained on Baby Yoda

(a) Testing Image 1: Ground Truth



(b) Testing Image 1: GP Colouring

(c) Testing Image 2: Ground Truth

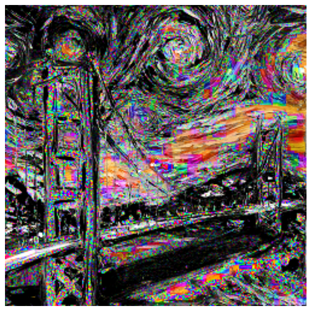
(d) Testing Image 2: GP Colouring

Fig. 10: GP colouring with pixel hue regression fitness and trained on Painting 1

(a) Testing Image 1: Ground Truth

(b) Testing Image 1: GP Colouring

(c) Testing Image 2: Ground Truth

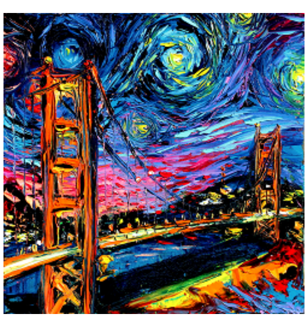


(d) Testing Image 2: GP Colouring

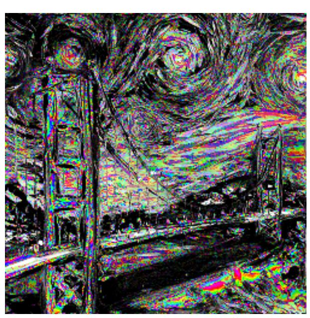
Fig. 11: GP colouring with colour distribution fitness and trained on Baby Yoda

(a) Testing Image 1: Ground Truth

(b) Testing Image 1: GP Colouring



(c) Testing Image 2: Ground Truth

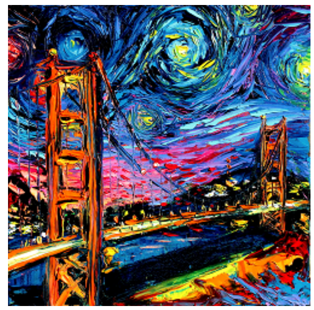


(d) Testing Image 2: GP Colouring

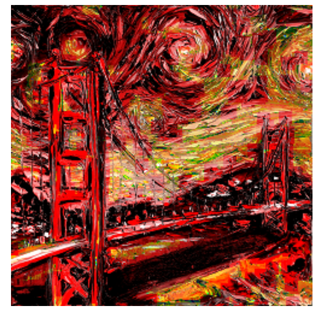
Fig. 12: GP colouring with colour distribution fitness and trained on Painting 1

(a) Testing Image 1: Ground Truth

(b) Testing Image 1: GP Colouring



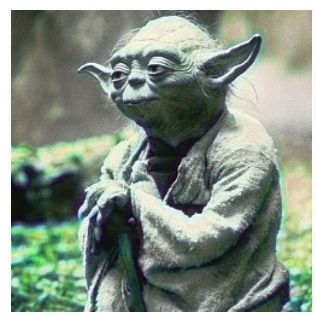
(c) Testing Image 2: Ground Truth



(d) Testing Image 2: GP Colouring

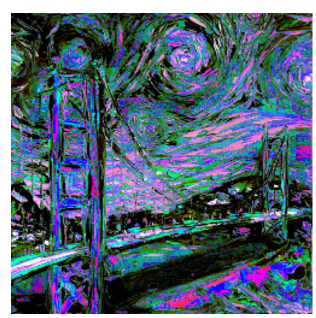
Fig. 13: GP colouring with hybrid fitness and trained on Baby Yoda

(a) Testing Image 1: Ground Truth



(b) Testing Image 1: GP Colouring

(c) Testing Image 2: Ground Truth



(d) Testing Image 2: GP Colouring

Fig. 14: GP colouring with hybrid fitness and trained on Painting 1

than with the regression. It also tends to choose colours that better correspond to the ground truth image albeit in the wrong locations.

## VII. Conclusion

Attempting to colour a greyscale image is a difficult task. Using GP to try and accomplish this task comes down to a good fitness function and the difficultly of the image. Three fitness functions have been tested in this paper but none of them have given acceptable results. However, many things have been learnt about GP's ability to evolve a solution to this problem and about the problem itself through our experiments.

Training and testing on images which have little difference between the colours yields better performance from the GP then training and testing on images which have many different colours. This is shown from the GP colouring examples for the different paintings, Yoda, and Baby Yoda images. When training with Baby Yoda and Testing on Yoda the images appear to be closely similar to the original colouring but, when training with painting 1 and testing on painting 2, the images do not appear to remotely match the original.

It should be noted again that the original saturation was used from the coloured image. Hence, we only evolved the hue of the images and then recoloured on top of the greyscale images. Due to this, there is no way to apply this GP to a pure greyscale image without the corresponding saturation values. If the findings from the GP were successful when trying to evolve the hue we were going to do the same thing with the saturation. If a new, more advanced fitness function is introduced to acceptably evolve the hue values of each pixel then it should also be used to analyze its performance on evolving the saturation. Once both have been evolved by the GP to an acceptable point they can be used in combination to colour greyscale images. A possible future experiment would be a co-evolution with hue and saturation.

It is again difficult to find an acceptable point because looking at how closely two images are to each other can, for the most part, be subjective. An important take away from these experiments would be that colouring bright and vibrant images is very difficult for GP to do, but if the target colour variety is low, then GP has a better chance of accurately colouring those greyscale images. If the colours are not the same in the training as they are in the testing, the colouring will turn out poorly for the GP based on our proposed fitness functions.

The three fitness functions used tended toward only using colours that were present in the training image. Therefore, if the training and testing images have different colours the chances the testing image will be coloured with colours it has but are not present in the training image are very slim or zero. In addition, trying to account for this problem by accounting for the distribution of colours to be consistent between the training images resulted in noisy performance. This is shown in our colour distributed images. Even though there may be a good amount of pixels that are within the same colour distribution in training, the GP could get a similar distribution by placing those coloured pixels anywhere on the target image which can lead to noisy images.

The pixel hue regression fitness seemed to only provide a colour filter on the entire image based on the most popular pixel hue during training. This leads to uninteresting colourings.

The colour distribution fitness subjectively did the worst overall. The colourings are very noisy and random. The GP seemed to be trying to get the colour variety with no regard for patterns or location. This makes sense given the fitness did not reward location.

The hybrid had interesting results with colourings that were more uniform but with more variety than the regression. It did not give a more accurate colouring than the regression for the low colour distribution image (Yoda) however, on the high colour distribution image (Painting 2) it seemed to subjectively perform the best under this fitness.

Even though the recolouring may not give an accurate colouring, there may be potential for artistic recolourings or image filters with this GP.

The GP may have generalized better on non-similar images if more training images with diverse colour palettes were used.

Overall, more research and investigation into this problem should be done with GP because it shows great promise given time and effort. One major and interesting application of using GP to colour greyscale images would be to fully recolour a black and white movie. Pointing out this interesting application might encourage others to try and find a solution to this problem.

## References


[1] George Lucas, director. *Star Wars Episode V: Empire Strikes Back*. 1980.
[2] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT PRes, 1992.
[3] Sean Luke et al. *ECJ 27: A Java-based Evolutionary Computation Research System*
[4] Machado et al. *Learning to Colour Greyscale Images*
[5] Rian Johnson, director. *Star Wars Episode VIII: The Last Jedi*. 2017.
[6] John Favreau, director. *The Mandalorian*. 2019.
[7] Aja Kusick, director. *Golden Gate Bridge*. nd.
[8] Aja Kusick, director. *Eiffel Tower*. nd.

## Appendix

### A. Best GP Trees

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=2557.023175902241 Adjusted=3.909268725242451E-4 Hits=15475
Tree 0:
 (/ 11x11_intensity_std_dev (+ 11x11_max_intensity
     (/ (/ 11x11_intensity_std_dev 11x11_max_intensity)
         (/ (/ (/ 11x11_intensity_std_dev 11x11_max_intensity)
             11x11_intensity_std_dev) (sqrt 11x11_intensity_std_dev)))))
```

Fig. 15: GP best tree for pixel hue regression fitness and Baby Yoda training set tested on Yoda

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=2448.046405597004 Adjusted=4.0832219337070093E-4 Hits=17118
Tree 0:
 (/ (/ (* 11x11_min_intensity 11x11_intensity_std_dev)
     11x11_max_intensity) (+ (+ 11x11_intensity_std_dev
     11x11_min_intensity) (/ (+ 11x11_intensity_std_dev
     (+ 11x11_intensity_std_dev (* 11x11_min_intensity
         11x11_intensity_std_dev))) 11x11_mean_intensity)))
```

Fig. 16: GP best tree for pixel hue regression fitness and Baby Yoda training set tested on Painting 2

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=9.51059544535636 Adjusted=0.09514208830497853 Hits=6
Tree 0:
 (square (+ (- (/ 11x11_intensity_std_dev
     (/ intensity 11x11_intensity_std_dev)) intensity)
     (* (/ (+ 11x11_mean_intensity 11x11_mean_intensity)
         (* intensity 11x11_min_intensity)) (1- intensity))))
```

Fig. 17: GP best tree for colour distribution fitness and Baby Yoda training set tested on Yoda

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=7.664940247468298 Adjusted=0.11540760483514906 Hits=8
Tree 0:
 (/ (* (sqrt (+ intensity 11x11_intensity_std_dev))
     (square (1- (- 11x11_min_intensity 11x11_intensity_std_dev))))
     (- 11x11_intensity_std_dev (* (square (-
         (- 11x11_intensity_std_dev 11x11_min_intensity)
         11x11_intensity_std_dev)) (- 11x11_intensity_std_dev
         11x11_min_intensity))))
```

Fig. 18: GP best tree for colour distribution fitness and Baby Yoda training set tested on Painting 2

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=12202.970676299865 Adjusted=8.194054431333582E-5 Hits=9843
Tree 0:
 (sqrt (sqrt (sqrt (/ (+ intensity intensity)
     (square (+ intensity 11x11_intensity_std_dev))))))
```

Fig. 19: GP best tree for pixel hue regression fitness and Baby Yoda training set tested on Yoda and Painting 2

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=7.692984967383569 Adjusted=0.11503528462916252 Hits=8
Tree 0:
 (sqrt (1- (/ (- 11x11_max_intensity intensity)
     11x11_intensity_std_dev)))
```

Fig. 20: GP best tree for colour distribution fitness and Painting 1 training set tested on Yoda

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=6.489964191871317 Adjusted=0.13351198675759715 Hits=9
Tree 0:
 (1- (/ (/ (- 11x11_mean_intensity intensity)
     (1- (/ 11x11_intensity_std_dev 11x11_mean_intensity)))
     11x11_mean_intensity))
```

Fig. 21: GP best tree for colour distribution fitness and Painting 1 training set tested on Painting 2

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=9.018134573716257 Adjusted=0.09981898253030225 Hits=10
Tree 0:
 (sqrt (/ (* (- intensity 11x11_mean_intensity)
     (sqrt 11x11_min_intensity)) (- (/ (* (- intensity
     11x11_mean_intensity) (square intensity))
     11x11_mean_intensity) (square intensity))))
```

Fig. 22: GP best tree for hybrid fitness and Baby Yoda training set tested on Yoda and Painting 2

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=14.92451557252249 Adjusted=0.06279625872736028 Hits=5
Tree 0:
 (sqrt (1- (sqrt (1- (/ 11x11_intensity_std_dev
     11x11_mean_intensity)))))
```

Fig. 23: GP best tree for hybrid fitness and Painting 1 training set tested on Yoda

```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=14.144231960753197 Adjusted=0.0660317408364805 Hits=7
Tree 0:
 (square (square (sqrt (sqrt (/ intensity
     11x11_max_intensity)))))
```

Fig. 24: GP best tree for hybrid fitness and Painting 1 training set tested on Painting 2