A Group Convolution Model for Hue-Invariant Classification

Felix O'Mahony [*1] **Miles Simpkins** [*2]

Abstract

Although several methods have previously been considered for the creation of roration-invariant image classifcation, colour-invariant image classifcation has not been explored. This paper introduces a color-invariant lifting layer to a group convolution binary classification problem. The results indicate that the model exhibits a strong capacity to predict the correct class from a set of images with varying colours, when only trained on a set of images with a single colour. Along with a presentation of the results and limitations of the model, this paper includes a proof that colour is a group, and original image sorting algorithms for dataset pre-processing.

1. Introduction

Group Convolutional Neural Networks (GCNNs) are able to sort visual content, recognize objects, and learn and interpret image patterns (Sharma, 2021) by utilizing the properties of groups. As a result, they are widely employed in the computer vision field with direct uses in facial recognition, medical image analysis, autonomous driving, among many others.

Intrinsically, GCNNs have a dependence on hue and typically these models are trained on large image datasets with a variety of hues to ensure accurate classification. Imagine a situation where the training dataset is limited to objects of a uniform hue that do not accurately describe the general population. In training the model on this limited dataset, poor performance in practice is expected. Instead of the conventional approach of expanding the dataset to include a variety of hues, this paper utilizes a colour invariant lifting layer to reduce the need for comprehensive training datasets.

Specifically, two group convolution models, one with and one without a colour invariant lifting layer, were trained to perform a binary classification problem: identifying cars and motorbikes. The models were trained only using red cars and motorbikes and their performance was evaluated using two datasets. The first dataset contains only red vehicles and the second contains vehicles of a variety of colours. The new lifting layer was shown to improve classification accuracy for the dataset containing all vehicles, while not harming performance in classifying red vehicles. This illustrates that the model exhibits a strong capacity to predict the class of images from a binary problem with varying colours, when the model is only trained on a single colour.

First, a background of group theory and group equivariant convolutional networks is presented. Next, the cars and motorbikes datasets are covered, along with their accompanying pre-processing sorting algoritms. This is followed by a proof that colour is a group. Then, a baseline classifier and the adaptations necessary for hue-invariant classification are discussed. Finally, detailed results are presented along with a discussion that includes limitations of the model and topics of future work.

2. Background

This section covers the base concepts utilized in this project: the definition of a group, and an introduction to group convolution neural networks for classification.

2.1. Group Theory

Since this report considers colour invariant classification with colour being defined as a group, it is necessary to formally define groups. A group is a set G alongside an operation that takes two elements of G and combines them to produce a third element of G (Yao & Corn). A group must satisfy the following set of axioms (Yao & Corn). For $x, y, z \in G$:

1. Associativity: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
2. Identity: There exists an $e \in G$ such that $e \cdot x = x \cdot e = x$ for any $x \in G$, where e is an identity element of G.
3. Inverse: For any $x \in G$, there exists a $y \in G$ such that $x \cdot y = y \cdot x$. y is an inverse of x.

[*]Equal contribution [1]University of Oxford, Oxford, United Kingdom [2]Princeton University, New Jersey, United States of America. Correspondence to: Felix O'Mahony <fo8087@princeton.edu>, Miles Simpkins <miless@princeton.edu>.

Proceedings of the International Conference on Deep Learning for Physical Systems, Copyright 2023 by the author(s).

4. Closure: The operation is defined as function which implies: For any $x, y \in G$, $x \cdot y$ is also in G.

In this project, groups refer to symmetry groups. Examples of symmetry groups are translations, rotations and reflections. Group p4, shown in Figure 1 includes translations and 90°rotations.

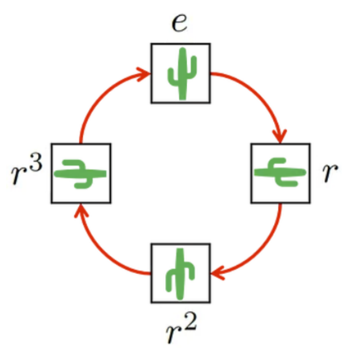

Figure 1. Group p4 describes 90°rotations (retrieved from (Engelenburg, 2020)).

2.2. Group Equivairant Convolutional Networks For Classification

Normal Convolution Neural Networks have a structure that preserves translation. This allows an object to be shifted along the sampling lattice without affecting the outcome of the network (Engelenburg, 2020). In other words, CNNs are translation equivariant: a convolution of a translated image is the same as translating the convolution of the original image. An example of consistent classification despite translated components is shown in Figure 2.

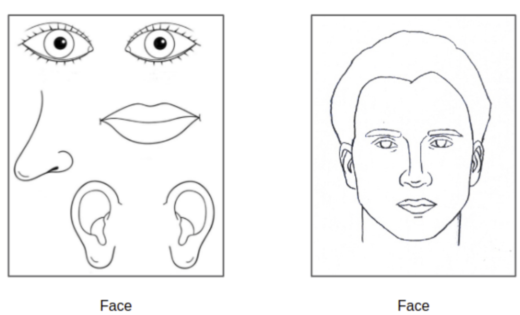

Figure 2. Translation Equivarince (retrieved from (Tan, 2021))

For image detection, this is an extremely valuable property that allows for improved classification performance compared to a normal neural network. Even with translation equivariance, CNNs must still be deep or wide to perform well.

Group Equivariant Convolutional Networks utilize symmetry group equivariance that can include rotations and reflections in addition to translations. This concept is illustrated in Figure 3. To be more precise, group equivariance means the feature map transforms the same as how the input transforms. Group invariance means the feature map doesn't change as the input changes. Although similar, the rest of this paper focuses on group invariance.

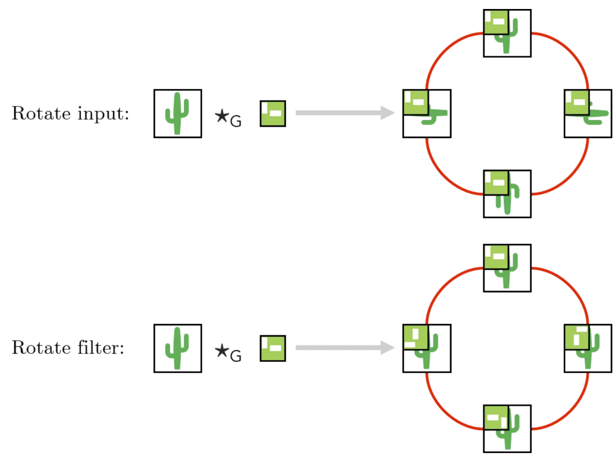

Figure 3. Group convolution of p4 on a planar input (retrieved from (Engelenburg, 2020)).

3. Dataset

In order to keep the problem simple and to focus on evaluating the colour invariant performance of the model, a binary classification problem was selected with the labels as car or motorbike. Two datasets were used: The Stanford Cars Dataset (Krause et al.) and the Motorbike Dataset (Images.cv). Sample images of this dataset are shown in Figure 4.

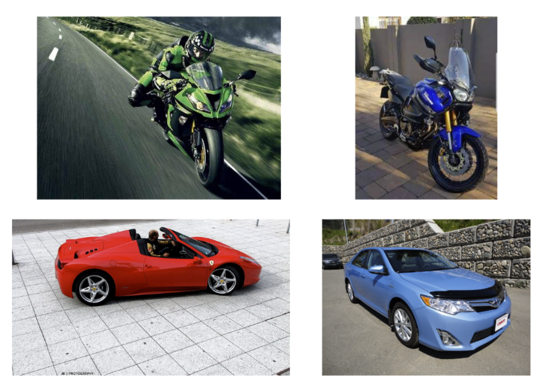

Figure 4. Sample images from the Stanford Cars and Motorbike datasets

The Stanford Cars Dataset is comprised of the exterior of 16,185 cars of varying points of view, colours, sizes, and backgrounds (Krause et al.). TThe Motorbike Dataset is comprised of 2,400 images of motorbikes with a similar variety of colour, sizes, points of view and backgrounds. The images of the cars and motorbikes are generally located towards the center of the image, not along the borders.

In order to test our hypothesis of colour invariant classification, it was necessary to train the model on objects of one colour. The target vehicle colour was selected as red for its contrast to the background. As a result, two sorting algorithms were produced to filter the vehicles: the first utilizes an HSL scoring system and the second sums the local softmax of the RGB values. The HSL scoring algorithm was selected for use.

3.1. HSL Scoring Algorithm

At a high level, the HSL scoring algorithm defines a score for each image, and evaluates each pixel of the image by increasing or penalizing the score depending on if the pixel has the desired hue, saturation and light. After all images are evalutaed, the scores are normalized by image size, sorted from most red to least red, and used as an input to the neural network.

Specifically, the desirable range of hue is set to be narrow, while saturation and light have larger acceptable values. The values used to sort the cars dataset are provided as an example in Table 1.

Parameter	Scale	Acceptable Range
Hue	0 to 360	$hue \leq 5$ OR $hue \geq 355$
Saturation	0 to 100	$sat \geq 70$
Light	0 to 100	$15 \leq light \leq 90$

Table 1. The parameter ranges that define the target red colour.

The scoring rules are shown in Table 2. In addition to receiving points for pixels in the target colour range, a scoring penalty was established for pixels that were too light or too dark regardless of hue. This assigns a greater weight to the value of red pixels.

Condition	Point Change
Target hue, sat, light	$+2$
Too light/dark	-1
Other	$+0$

Table 2. HSL Scoring Algorithm reward and penalty rules

To summarize, the steps of the algorithm are listed below:

1. Convert each image from RGB (red, green, blue) to HSL (hue, saturation, light).
2. Remove the border.
3. For each image, assign a "redness score".
4. Go pixel by pixel to assess the HSL value and update the image score.
5. Sort images from highest to lowest normalized score.

3.2. RGB Softmax Algorithm

Although this algorithm was not used in the final sorting of the vehicles, it will be covered for completeness. At a high-level, this algorithm uses the relative RGB magnitudes of each pixel in an image to determine redness. This approach is taken because pixels that appear red in colour generally have higher R values compared to G and B values. To quantify the relative differences between RGB values, a local softmax was taken between the RGB values of each pixel. Given an RGB vector $\hat{z}$:

$$\hat{z} = < c_r, c_g, c_b > \tag{1}$$

the softmax with respect to the r value is:

$$\sigma(\hat{z}_1) = \sigma(c_r) = \frac{e^{c_r}}{e^{c_r} + e^{c_g} + e^{c_b}} \tag{2}$$

After each local softmax has been computed, the sum of the red softmax values is computed and represents the image redness. The image redness values are then sorted. The algorithm is summarized below:

1. Leave the images in RGB.
2. Take the softmax of the RGB values of each pixel.
3. Sum the red values of each softmax across each image.
4. Sort images from highest to lowest normalized "redness".

This sorting algorithm was not selected for use due to a few reasons. First, the softmax operation depends on relative differences and by itself cannot guarantee that a pixel appears red. For example, consider Figure 5 which shows a single pixel with $< c_r, c_g, c_b > = < 30, 1, 1 >$ displaying a dark colour.

Equation 3 reveals that a high softmax value for the R pixel occurs despite the box not appearing as red.

$$\sigma(< 30, 1, 1 >) = < 0.999,\ 2.51 \cdot 10^{-13},\ 2.51 \cdot 10^{-13} > \tag{3}$$

Figure 5. A dark pixel with an RGB value of $< 30, 1, 1 >$.

3.3. Sorting Algorithm Results

Once the images were sorted, they were resized to be 128 x 128 pixels for input to the model. As an example, the most red and least red car of the sorted dataset are shown in Figure 6.

Figure 6. The most and least red cars sorted using the HSL Scoring Algorithm.

4. Literature Review

We begin by introducing two previous ideas which are connected to the research undertaken as part of this project. Group convolution is a widely explored topic, and several examples of this methodology applied to other types of invariant problem exist. Secondly, we introduce pre-existing methodologies for colour invariant classification.

4.1. Group Convolution for Invariance

Several methods have previously been considered for the creation of rotation-invariant image classification. Chidester et al (Chidester et al., 2018), propose rotation-invariant CNN (RiCNN). The method achieves its rotation invariance through the application of a set of rotated convolution filters across the input image. A fourier transform is taken of the final output of the convolutional layers which produces a cyclic-shift space of the rotations of the feature maps. This final feature space is thereby invariant under rotation, which can then be applied to a multi-layer perceptron network to perform classification.

Follman et al (Follmann & Bottger, 2018) implements rotation invariant classification by incorporating rotation invariance into the convolutional layers of a classification network explicitly. Their results show that while training only on unrotated examples of data from the MNIST dataset, the model can classify images at any rotation with greater accuracy than a model which does not incorporate rotationally-invariant convolution.

4.2. Colour Invariant Classification

While no research currently exists regarding hue invariant colour classification, prior research does exist regarding invariant classification of some colour properties. Chong et al (Chong et al., 2008) achieves good results for luminance invariant classification by deriving a new colour space which is robust to changes in the illumination of the pixel value.

5. Colour as a Group

While this model implements invariance, rather than equivariance, it is still of interest to demonstrate that the hue of a pixel represents a group under linear changes to its value. Given a pixel c in an image, we assign the variable $\mathbf{c}_{RGB} = [c_r, c_g, c_b]$ to the RGB value of the pixel. We assign the variable $\mathbf{c}_{HSL} = [c_H, c_S, c_L]$ to the same pixel value represented in the HSL colour space. Let the transform $T_{RGB \to HSL}(c_{RGB}) : \mathbb{R}^3 \to \mathbb{R}^3$ represent the mapping of the pixel colour from the RGB to HSL space, with $T_{HSL \to RGB}(c_{HSL})$ representing the inverse transformation. We apply the operation of our group convolution classifier to pixels as they exist in the RGB space. However, it is convenient to demonstrate that hue is a group in the HSL space. First, a proof is given that c_{HSL} is a group under a linear additive change to hue value. Secondly, a proof is given that the mapping $T_{RGB \to HSL}(c_{RGB})$ is bijective. That is to say that the equivalent inverse transformation $T_{HSL \to RGB}(c_{HSL})$ exists and is unique for all cases except where $c_R = c_G = r_B$, for which special consideration is given at the end of this section.

We note that typically in image representation, colour values $c_r, c_g, c_b, c_H, c_S, c_L$ are typically integers in $[0, 255]$ rather than real numbers. However, the approximation taken here that they may be treated is reasonably valid since there is reasonably fine control of image colour through the variation of these channel values. As such, the impact of discretisation is minimal.

5.1. HSL as a Group

Here we prove that the hue of a colour specified by $\mathbf{c}_{HSL}$ is invariant under linear additive hue change. The operation may be specified mathematically by the addition of a vector Δc_H, so that $c'_H = \mod(c_H + \Delta c_H, 360°)$. The modulo operation means that the set of hue values is a circular shift space.

A group, as defined earlier, must have five properties:

- Closure
- Associative
- Identity element
- Inverse element
- Commutative

5.1.1. Closure

The set of hue values is constrained to the domain $[0, 360)$. Although addition would mean that the hue value could exceed this, the implementation of a modulo operation reforms the value within the original domain. Therefore the hue is closed under the operation.

5.1.2. Associativity

The combined function of hue adaption under two shifts $\Delta c_H^{(1)}$ and $\Delta c_H^{(2)}$ may be represented by

$$c'_H = \mod(\Delta c_H^{(2)} + \mod(\Delta c_H^{(1)} + c_H)) \tag{4}$$

It is sufficient to note that, since the modulo function makes hue a circular shift space, that applying the function twice with an intermediary additive step is equivalent to applying it once around the entire expression:

$$c'_H = \mod(\Delta c_H^{(2)} + \Delta c_H^{(1)} + c_H) \tag{5}$$

Which is evidently associative since addition is associative.

5.1.3. Identity

The identity element is simply $\Delta c_H = 0$. Noting that originally $c_H \in [0, 360)$, by definition of the modulo function:

$$c'_H = \mod(\Delta c_H + c_H) = c_H \tag{6}$$

5.1.4. Inverse Element

Applying equation 5 where $\Delta c_H^{(1)} = -\Delta c_H^{(2)}$:

$$c'_H = \mod(\Delta c_H^{(2)} - \Delta c_H^{(2)} + c_H) \tag{7}$$
$$c'_H = \mod(c_H) \tag{8}$$
$$c'_H = c_H \tag{9}$$

5.1.5. Commutative

Again applying equation 5, it is self evident that the operation is commutative since addition is commutative.

Hue is thereby a group under linear additive value shift.

5.2. RGB to HSL as a Bijective Mapping

The transformation $T_{RGB \to HSL}$ which maps a colour from its representation in the RGB space to the HSL space when it is not the case that $c_r = c_g = g_b$ is given by the equation from (RapidTables, 2023):

$$c_{\text{max}} = \max(c_r, c_g, c_g) \tag{10}$$
$$c_{\text{min}} = \min(c_r, c_g, c_g) \tag{11}$$

$$c_H = \begin{cases} 60° \frac{c_g - c_b}{c_{\text{max}} - c_{\text{min}}} & \text{if} \quad c_{\text{max}} = c_r \\ 60° \frac{c_b - c_r}{c_{\text{max}} - c_{\text{min}}} + 120° & \text{if} \quad c_{\text{max}} = c_g \\ 60° \frac{c_r - c_g}{c_{\text{max}} - c_{\text{min}}} + 240° & \text{if} \quad c_{\text{max}} = c_b \end{cases} \tag{12}$$

$$c_S = \frac{c_{\text{max}} - c_{\text{min}}}{1 - |2c_L - 1|} \tag{13}$$

$$c_L = c_{\text{max}} + c_{\text{min}}/2 \tag{14}$$

Physically, this transform represents a mapping from representation of colour on a cube (with axes c_r, c_g, c_b) to a set of axes perpendicular to the diametric space diagonal of the cube. This process is illustrated in figure 7.

This projection illustrates the bijectivity of the mapping, with hue indicating the angle around teh diametric space diagonal, the radius indicating the saturation, and the extent along the diagonal indicating the luminosity. A complete formula for the inverse transform is given in (RapidTables, 2023).

Since the mapping is bijective, and the hue transformation of colour in the HSL colour space is a group, hue transformation of colour in the RGB space is also a group. $\square$

Some extra consideration must be given to the case where $c_r = c_g = c_b$. This is set of greyscale colours, and so hue transformations have no impact on the overall colour. As such, the hue assigned to the colour is inconsequential, since the inverse transform will always simply return the original values.

6. Classifier

We begin by describing the baseline classification model used, then by describing the adaptions made to the model to create a hue-invariant classification.

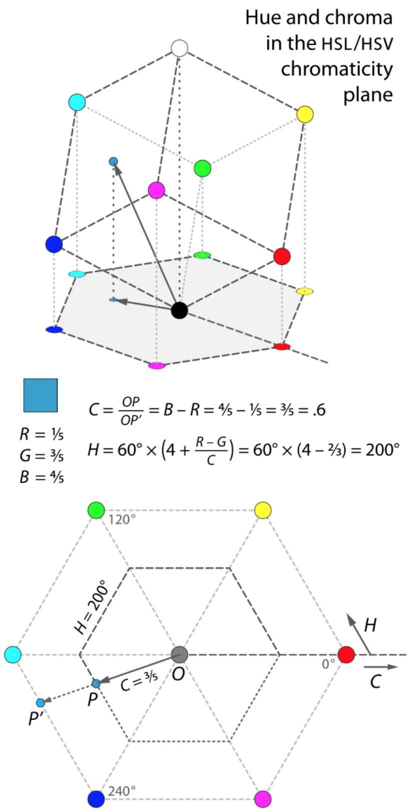

Figure 7. Process of mapping colour from RGB to HSL. A projection of the RGB colour sphere is taken in a direction orthogonal to its diametric space diagonal. Image from (Rus, 2010)

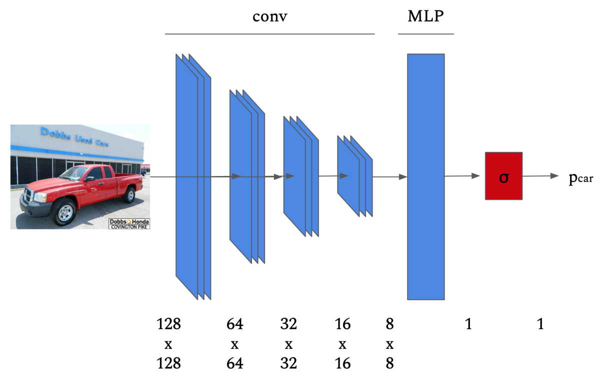

Figure 8. Baseline classification model. Four convolutional layers followed by batch norm 2D layers and ReLU activation functions shrink the image to an (8×8) feature map. At this stage, the feature map is flattened and input to a multi-layer perceptron. The MLP consists of three layers, the first two of which use ReLU activation functions. These layers further reduce the dimension of the signal to a single value. The final activation function of the network is a sigmoid. This constrains the output to $(0, 1)$, representing the probability that the image is a car.

6.1. Baseline

Serving as a baseline for the model, a simple standard convolution model was implemented in line with convolutional classification models described in section 2.2. The model is illustrated in figure 8. The model is formed of 4 convolutional layers. Each layer carries a kernel size of (4×4) and a stride of 2. Consequently, at each stage of the convolution the image dimension shrinks by two. Though not illustrated, the number of channels is 3 at each stage until the last, when the number of channels is reduced to 1.

This output is then flattened to a vector of length 64, and used as the input to a multi-layer perceptron network. This network has three layers. The first two use ReLU activation functions. The output of the network, a single value, is used as the input to a sigmoid activation function which returns the probability that the input image was a car rather than a motorbike.

6.2. Training

The network was trained according to a mean squared error loss function between the network output $y_{\text{predicted}}$ and the target label y_{target}:

$$\mathcal{L} = \frac{1}{N}(y_{\text{target}} - y_{\text{predicted}})^2 \tag{15}$$

A full description of training is given in appendix section A.

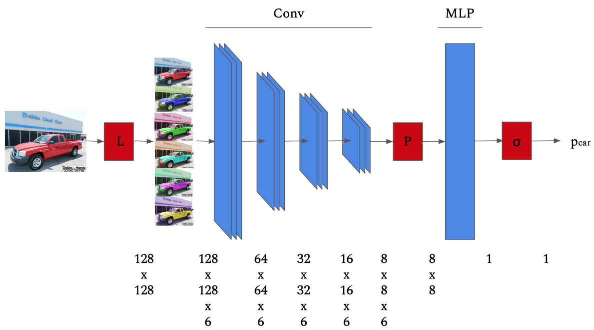

Figure 9. Modified convolutional network. Now, a lifting layer (L) modifies the network to give a further dimension with 6 layers by replicating the image at different hue values. A max pooling layer (P) reduces these back to 1 by taking the maximum value across the layers. This introduces hue invariance.

6.3. Colour Invariant

Two adaptions are made to the baseline network to introduce colour invariance. The modified network is shown in figure 9. Firstly, the input image is lifted by a layer introduced before the convolutional layers. This lifting layer modifies the input image to produce six iterations. Each iteration adjusts the hue of the pixels in the input image by $30°$ compared to the last.

This means that the inputs to the convolution layer now have an additional dimension of size 6. The convolution layers are applied to each separate image iteration as if it were the original image used in the baseline model. As such, the output from the final convolutional layer has one additional dimension, also of size 6. Returning the image to the same size as the baseline model, a max pooling layer is used across all six layers.

This arrangement introduces a strict invariance in the output (8×8) feature map of the convolutional layers across $30°$ hue changes. This is because each of the six colour variants is treated identically by the convolutional network. At the output stage, the feature map representation with the highest value is selected to pass its value to the MLP. Were the image hue shifted by $n \times 30°, \quad n \in \mathbb{Z}$, one of the other convolutional outputs would have the highest value, but that highest value would remain the same.

To illustrate this principle, a sample image drawn at random from the training dataset is input to the network and the (8×8) feature maps output by the final convolutional are illustrated as colour maps. These are compared between the baseline and modified networks in figure 10. Clearly, when the colour invariant network is used, the feature map remains the same regardless of the image hue. By contrast, when the baseline model is used the feature map changes significantly as the hue changes.

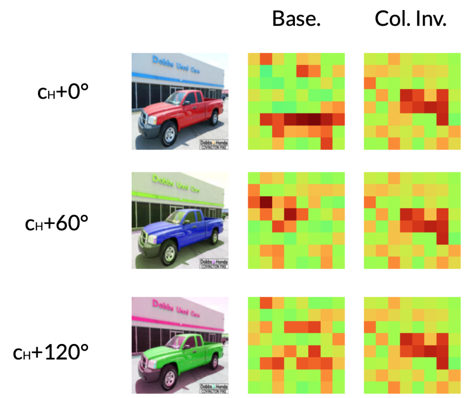

Figure 10. Feature maps output by the convolutional layers as input image hue changes. When the colour invariant network is used, the feature map does not change. This is not the case for the baseline model. The feature maps are illustrated here as pixel grids, with each pixel colour indicating value with a colour scale ranging between [0,1].

Table 3. Classification accuracy (↑) of the baseline model and colour invariant model on a dataset of images containing only red vehicles and one of images containing vehicles of all colours.

Model	Red Vehicles (%)	All Vehicles (%)
Baseline	83	67
Colour	83	83

As before, this network was trained according to a mean-squared error loss.

7. Results

To test the capacity of the modified network to successfully classify objects regardless of their colour, both networks were trained on a dataset which contained only red vehicles.

The trained models were then used to classify two validation datasets. The first dataset was partitioned from the same set of images from which the training dataset was taken, although the two datasets were selected so that there was no intersection between the two. This dataset also only contained red vehicles. The second dataset was selected at random from the original dataset, and so contained vehicles of all colours.

The resulting classification accuracy of both models on the two datasets is shown in table 3. The mean squared error of the two models on the validation datasets is shown in table 4.

The model which introduces colour invariance exhibits an improved performance over the baseline model in terms of classification accuracy and mean squared error.

Table 4. Mean squared error (↓) of the baseline model and colour invariant model on a dataset of images containing only red vehicles and one of images containing vehicles of all colours.

Model	Red Vehicles (σ)	All Vehicles (σ)
Baseline	0.12 (0.24)	0.24 (0.32)
Colour	0.14 (0.29)	0.13 (0.29)

8. Discussion

The results of this report are promising, and illustrate that the model, as proposed, exhibits a strong capacity to predict the class of images for a binary problem with varying colours, even if the model is only trained on a single colour. However, there are some limitations of the work as presented, which would need to be considered in the context of a broader application of this model. There are also several possible routes for improvements to the model which should be considered.

8.1. Results

The classification accuracy of the colour invariant model and the baseline model are identical for the case that they are tested within the domain of their training data, that is among red vehicles. This demonstrates that the lifting arrangement we introduce does not in any way degrade the normal performance of the network.

However, once the validation dataset is expanded to include a range of colours, the classification accuracy of the baseline model reduces by 16% to 67%. This drop is significant, and this classification accuracy is now very low (given that a random classifier would expect a classification accuracy of 50%). When the network is modified as described in this paper, however, the classification accuracy remains the same once the validation dataset is expanded to all colours. This demonstrates the effectiveness of our model for the classification of datapoints of any hue, even when the training dataset contains a limited number of hues.

The data from the mean squared error shows similar results. Once colour invariance is introduced, the mean squared error remains the same between the two different datasets. One interesting point of note is that the standard deviation of the mean squared error is higher when the colour invariance is introduced. This might indicate that the colour invariant network makes less confident predictions about the class of a vehicle, even if its classification accuracy is the same as the baseline when tested across red vehicles.

8.2. Limitations

The first limitation of our model is the additional time taken to train the network. This limitation is small, since the majority of the time taken to train a model consists of the time taken to preform backpropagation. However, the additional computation required to lift and perform a forward pass with the additional images increased the time to train an equal number of epochs by 18%.

A second limitation is that the model as introduced would be inappropriate in several important scenarios. For example, in cases where we might want to classify a vehicle with more detail than simply its type. In this case, we might hope that a network is able to classify objects as 'red' and 'blue' vehicles, rather than simply as vehicles. The network as proposed would be inappropriate for this task. Further, there are several examples of objects whose colour is significant, such as flags or flowers. In both of these examples, introducing colour invariance could reduce the capacity of the network to identify objects, since it would confuse multiple objects of identical class.

8.3. Future Work

There is ample scope for future work in this regard. One option would be to expand the scope of testing. For example, one could train both models on images of both classes of image with no colour constraint. In this case, there would be interest in identifying whether the colour invariant network was also able to perform more accurate classification.

Another addition would be to consider whether other features of image colour, such as saturation or brightness, could be manipulated in the same way. This could give classification networks a greater robustness to effects like variable lighting across input images.

References

Chidester, B., Do, M. N., and Ma, J. Rotation equivariance and invariance in convolutional neural networks, 2018.

Chong, H. Y., Gortler, S. J., and Zickler, T. A perception-based color space for illumination-invariant image processing. *ACM Trans. Graph.*, 27(3):1–7, aug 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360660. URL https://doi.org/10.1145/1360612.1360660.

Engelenburg, C. v. Geometric deep learning: Group equivariant convolutional networks. *Medium*, May 2020. URL https://medium.com/swlh/geometric-deep-learning-group-equivariant-convolu

Follmann, P. and Bottger, T. A rotationally-invariant convolution module by feature map back-rotation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 784–792, 2018. doi: 10.1109/WACV.2018.00091.

Images.cv. Search and downlod labeled computer vision image datasets. URL https://images.cv/dataset/motorbike-image-classification-dataset.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. URL https://www.cv-foundation.org/openaccess/content_iccv_workshops_2013/W19/html/Krause_3D_Object_Representations_2013_ICCV_paper.html.

RapidTables. Rgb to hsl color, 2023. URL https://www.rapidtables.com/convert/color/.

Rus, J. Hsl-hsv hue and chroma, 2010. URL https://commons.wikimedia.org/wiki/File:HSL-HSV_hue_and_chroma.svg.

Sharma, P. Applications of convolutional neural networks(cnn), Oct 2021. URL https://www.analyticsvidhya.com/blog/2021/10/applications-of-convolutional-neural-networks

Tan, K. Capsule networks explained, Jan 2021. URL https://kndrck.co/posts/capsule_networks_explained/.

Yao, B. and Corn, P. Group theory. *Brilliant.org*. URL https://brilliant.org/wiki/group-theory-introduction/.

A. Training

Training was conducted over 500 epochs. Data in the dataset of red vehicles was split between the training and validation dataset according to an 80/20 ratio. A learning rate of 0.001 was used, as this was found to give optimal results. The loss curves over training for the models used in this report are shown in figure 11.

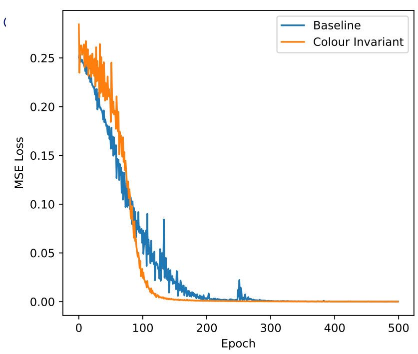

Figure 11. MSE loss curves of both the baseline and colour invariant model over the course of 500 training epochs.