# [INFO8010] Colorful image colourisation of cat images

**Bernard Aurélien**,[1] **Fery Loïs**,[2] and **Courtoy Boris**[3]

[1] *aurelien.bernard@student.uliege.be (s176639)*
[2] *lois.fery@student.uliege.be (s175043)*
[3] *boris.courtoy@student.uliege.be (s175068)*

## I. INTRODUCTION

Given a grey-scale image, being able to attribute a colour to each pixel in order to make a realistic and visually appealing result is a hard task. Performing this task requires the recognition of context of the image at each pixel, detecting the boundary of images, and furthermore, it requires an understanding of what colours are admissible for each object. A network that is able to perform this task in a satisfactory manner would exhibit all these properties that are central to computer vision. Before the discovery of transformers and their application to visual tasks[1], Convolutional Neural Networks(CNNs) were the leading architectures for computer vision related tasks. These architectures are still widely used today, both on their own and in conjunction with transformers [2].

In this project, we give a new perspective on the method described in the paper by Zhang et al. [3]. In their work, a CNN together with a tailor made loss function for the task are proposed to give realistic and colourful colourisation of greyscale images. To emphazise the multi-modality[4] of the problem, the network is not trying to recover the ground truth colourisation, but just one plausible colourisation for the grey-scale image. In their work, they do not constrain the type of image used in their network. We decided to constrain the type of images used in our datasets to cat images only. By constraining the type of images used, we investigated if this would improve the result of the colourisation, in order to have a specialised model for the recolourisation of only a certain type of pictures. Designing such a model would also suggest that CNNs are sufficient to attain automatic colourisation of images that make their colours indistinguishable of ground truth.

## II. RELATED WORK

Previous work to Zhang's contribution consisted mainly in learning a regression model to infer the colour of pixels from a continuous colour space [5]. Other work consisted in classifying the colour of each pixel to a quatized set of colours [6]. But none of these methods leverage the same amount of data or network size than what was seen in Zhang's paper.

Concurrently to Zhang et al., other research was conducted to use large scale networks and data sets [7][8]. However, these papers did not use the same loss function and colour re-balancing as the one used by Zhang, which is tailored to obtain meaningful colourisations. Indeed, these colourisations yielded less appealing results showing sepia or discolourisation effects. An explanation for these results will be discussed later in the loss function section.

Later approaches to recolourisation have used Generative Adversarial Networks [9]. However, this technique needs large amounts of data and is sensitive to unbalanced datasets where one type of scene is more present than the others. The results are colourful and mostly realistic. Nonetheless, in large datasets with a wide variety of scenes, miscolourisations are still common. Finally, there has been work using self attention to recolorize images [10]. This work has achieved colourisations that are often indistinguishable or preferred to ground truth images by human evaluators.

## III. METHODS

### A. Estimating probability instead of colours

The main idea of Zhang et al. that was reused for this project was to work with probability distribution over quantized pixel colour instead of direct pixel colour classification. This makes the prediction and network training more efficient due to the finite nature of the quantization. Also, a probability distribution is a more natural way of picturing colourisation as a same object can have several different colours depending on the instance. This can easily be seen with apples as an example. Some apples are green, some are red, but very rarely an apple is black or brown.

As we work in the LAB colour space, the idea is to discretise the AB value space into bins with a given size. In our application bins of $10 \times 10$ were used. Each bin covers different colours, but only the colour at the center of the bin was kept to represent it. We thus represent one colour per bin. From this grid and the colours met in the training set, we can represent a set of Q colours which is the set of colours in our gamut. Our gamut is found by selecting all the bins that were not empty after mapping all the colours found in the dataset to the discretised AB colour space.

### B. Soft encoding of colour probability

To convert a pixel colour (from a picture) to a colour probability distribution a soft-encoding scheme was used. This consists in first finding the n-nearest neighbours of the pixel colour among the computed in-gamut colours. Then, they are weighted proportionally to their distance to the true pixel colour. With one-hot encoding, only the nearest discretised colour takes 100% of the weight. Soft encoding allows for a smoother and more efficient training.

In our case we took $n = 5$ and the weighting was done using an operation equivalent to applying a convolution with a Gaussian kernel of variance $\sigma = 5$ on the 1 hot encoding AB space. Also, in order to improve training time, the result of the soft encoding for each image was kept in a hash-table. This was done because the soft encoding is necessary for the loss computation at training. As this operation is computationally expensive, computing it once and saving it for the rest of the training saved substantial training time.

### C. Network architecture

For this application it was decided to use the same network as the one proposed in the original paper. This network takes as input a matrix of size $(1 \times H \times W)$ that is the lightness channel of the lab images. The output is a tensor of Q channels and one fourth of the original size $(Q \times H/4 \times W/4)$. Each one of the Q channels contains a discrete probability distribution over the Q colour of the gamut. This gives a probability for each colour to each pixel of the image.

The convolutional neural network is composed of 8 different convolution blocks totalling 22 convolutions. These blocks are described in figure 1. The 0.5 stride in layer 8_1 is a 2× upscale of the resolution achieved with a transposed convolution layer. Every other up-scaling was done using bi-linear interpolation.

| Layer | Resolution out | Channels out | Stride | Kernel dilation | Accumulated downscale |
|---|---|---|---|---|---|
| Lightness | 224 | 1 | – | – | – |
| conv1_1 | 224 | 64 | 1 | 1 | 1 |
| conv1_2 | 224 | 64 | 2 | 1 | 1 |
| conv2_1 | 112 | 128 | 1 | 2 | 2 |
| conv2_1 | 112 | 128 | 2 | 1 | 2 |
| conv3_1 | 56 | 256 | 1 | 1 | 4 |
| conv3_2 | 56 | 256 | 1 | 1 | 4 |
| conv3_3 | 56 | 256 | 2 | 1 | 4 |
| conv4_1 | 28 | 512 | 1 | 1 | 8 |
| conv4_2 | 28 | 512 | 1 | 1 | 8 |
| conv4_3 | 28 | 512 | 1 | 1 | 8 |
| conv5_1 | 28 | 512 | 1 | 2 | 8 |
| conv5_2 | 28 | 512 | 1 | 2 | 8 |
| conv5_3 | 28 | 512 | 1 | 2 | 8 |
| conv6_1 | 28 | 512 | 1 | 2 | 8 |
| conv6_2 | 28 | 512 | 1 | 2 | 8 |
| conv6_3 | 28 | 512 | 1 | 2 | 8 |
| conv7_1 | 28 | 256 | 1 | 1 | 8 |
| conv7_2 | 28 | 256 | 1 | 1 | 8 |
| conv7_3 | 28 | 256 | 1 | 1 | 8 |
| conv8_1 | 56 | 128 | .5 | 1 | 4 |
| conv8_2 | 56 | 128 | 1 | 1 | 4 |
| conv8_3 | 56 | 128 | 1 | 1 | 4 |

FIG. 1. Table describing the architecture of the network. Taken from the original paper by Zhang et al. Every block ends with a batch normalisation layer except the last one. The last one ends with a $(1 \times 1)$ kernel convolution and a softmax on the channels' dimension. This is to normalise the probability distribution over the Q colour bins.

### D. Loss function

The classical loss function for this type of setting is the Euclidean distance loss between predicted and ground truth colours. However, this metric doesn't suit the task as the image colouring problem is multi-modal. Using it would result in an averaging effect. In order to reduce the average loss, the trained network would be likely to produce a greyish and desaturated colourisation that is in between the observed colours.

Instead, we used the loss function introduced by Zhang to compare the predicted probability distribution over possible colour against ground truth distribution. The multinomial cross entropy loss function that was used can be seen in equation 1. The ground truth distribution being the one created using the soft-encoding scheme.

$$\mathbf{L}_{cl}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\sum_{h,w} v\left(\mathbf{Z}_{h,w}\right) \sum_{q} \mathbf{Z}_{h,w,q} \log\left(\widehat{\mathbf{Z}}_{h,w,q}\right) \quad (1)$$

Where $\mathbf{Z} \in [0,1]^{H \times W \times Q}$ is a probability distribution over possible colours, and $v(\cdot)$ is a weighting term that is used for class re-balancing. This re-balancing term gives a greater weight to colours that are more rare. This counteracts the averaging that is present in the Euclidean distance loss.

$v(\cdot)$ is computed by taking the empirical distribution of colours over the full dataset $\widetilde{\mathbf{p}}$ and reweighting the probability distribution using a factor $\lambda \in [0,1]$. The analytical equations implemented for this can be seen in equations 2. The value of $\lambda$ used in our implementation was $\lambda = 0.5$. This choice was made to be consistent with the original paper and have results that can be compared.

$$v\left(\mathbf{Z}_{h,w}\right) = \mathbf{w}_{q^*}, \text{ where } q^* = \arg\max_q \mathbf{Z}_{h,w,q}$$
$$\mathbf{w} \propto \left((1-\lambda)\widetilde{\mathbf{p}} + \frac{\lambda}{Q}\right)^{-1}, \quad \mathbb{E}[\mathbf{w}] = \sum_q \widetilde{\mathbf{p}}_q \mathbf{w}_q = 1 \tag{2}$$

Fundamentaly, this loss function has two main advantages: It compares pixels in terms of probability distributions, and it counteracts the averaging effect. This achieves smooth and efficient training while yielding vibrant colours.

### E. Training

The dataset that we used to train the network was the "Dogs Vs Cats" kaggle dataset[11]. More precisely, we used approximately 4000 cat images as training set. The idea, as mentioned previously, was to constraint the recolouring problem to only one type of scene and thus to achieve good result even when using a small training set.

The network was trained over 300 epochs on a "GeForce RTX 2060 SUPER" GPU. At each epoch, batches of 32 random images were taken. The first step of the training was to split the lightness channel from the AB channels and feed it to the CNN to get a colour probability distribution for each pixel at the output. Then, the soft encoded distribution for the input images was retrieved from the precomputed hash table. The inference loss was computed using the previously explained multinomial crossentropy loss. It compared the estimated distribution to the soft encoded distribution. Finally, the loss was back-propagated through the network and the weights were updated using the Adam optimiser with its default beta parameters : $\beta_1 = 0.9$, and $\beta_2 = 0.999$ and an initial learning rate of $1 \times 10^{-4}$. The recorded training losses can be seen in figure 2. Test losses were not recorded as they were not relevant to the task. Good colourisations can be achieved while not being accurate compared to the ground truth. The results of training on test cases were evaluated with a qualitative analysis. This analysis is discussed in section V.

## IV. FROM PROBABILITY DISTRIBUTIONS TO COLOURS

To estimate a pixel colour from a probability distribution, we interpolate by re-adjusting the temperature T of the softmax distribution. Then we take the average of the result. The performed operations can be seen in equations 3 and 4

$$\mathcal{H}\left(\mathbf{Z}_{h,w}\right) = \mathbb{E}\left[f_T\left(\mathbf{Z}_{h,w}\right)\right] \tag{3}$$

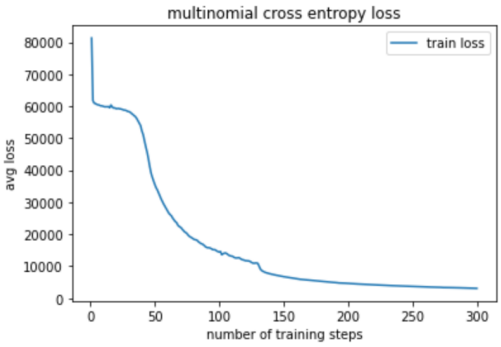


FIG. 2. Graph of the training of the model on 4000 cats pictures of the "Dogs Vs Cats" kaggle dataset. The loss correspond to the loss of a 32 pictures batch.

$$f_T(\mathbf{z}) = \frac{\exp(\log(\mathbf{z})/T)}{\sum_q \exp\left(\log\left(\mathbf{z}_q\right)/T\right)} \tag{4}$$

Increasing T yields a colourisation that is closer to the averaging of all probabilities. This yields colours that are less saturated but more spatially consistent. Decreasing T yields more vibrant colours by making the distribution tighter around its maximums. This improves vibrancy at the cost of spacial consistency. A good compromise was found by the authors of the original paper around the values of $T = 0.38$. This value was selected to be able to better compare our results.

## V. RESULTS

### A. Qualitative analysis

From figures 3 and 4, we notice that our trained model achieved good performance on the training set. There are nonetheless some inconsistencies. For example, the patches of the basket or the pink pattern that can be seen on the wall of picture 3.

Concerning the testing set, we can see in figure 4 that our model generalised poorly. Indeed, we can see blue and brown stain that are not consistent with the context. Also, pictures are greyish overall.

### B. Quantitative analysis

As mentioned earlier, the performance of an image recolouring model is very difficult to measure quantitatively. Indeed, the model's goal is to recolour picture in a way that could potentially fool a human observer. This criterion is difficult to evaluate with a function. The evaluation method that is proposed in the paper is a "Colourisation Turing test" that would have been performed by asking people to guess if a picture has been

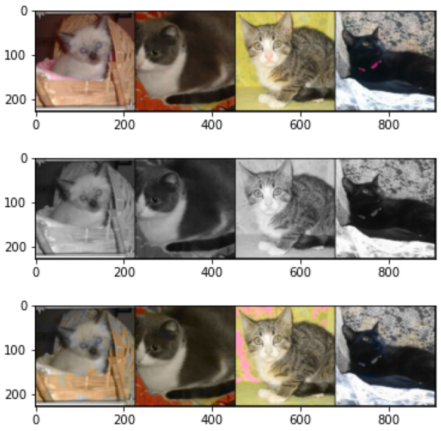

FIG. 3. Colourisation on the cat training set. The first, second and third row are respectively: truth pictures, grayscale images, and recoloured versions

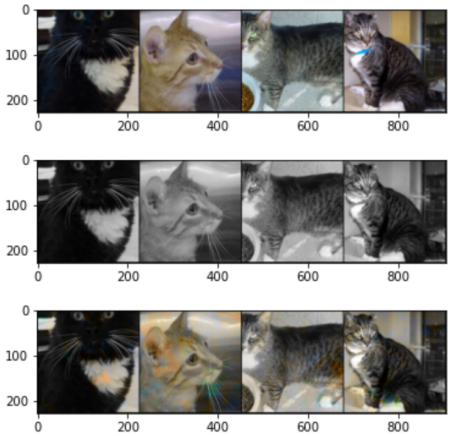

FIG. 4. Colourisation on the cat testing set. The first, second and third row are respectively: truth pictures, grayscale images, and recoloured versions

recoloured or not. A convincing relcolourisation would achieve a 50% chance of fooling an observer.

## VI. DISCUSSION

From what can be qualitatively observed, our image recolouring implementation is functional. Our model is able to re-colourise correctly the pictures of the training set. However, we can see that it failed to generalise to new pictures.

This can perhaps be explained by the limited amount of pictures that were used to train the model, or the fact that cats pictures often shares the same kind of colour. A training set showing different variety of colours could have been more suitable.

Another cause could be that we have trained the network for too many epochs, and that we over-fitted the model with the training data. Or on the contrary, we might have under-fit the model by making too few epochs. Indeed, the loss was non zero at the last loss plateau during training. The training could have still been at the beginning of the double descent curve. In this case more epochs would have improved the generalisation.

In image recolouring, finding a good stopping criterion is difficult. This is due to the lack of a good evaluation criterion for the model on the test set. This lead us to have good qualitative results on training data, but not on test data.

### A. Further considerations and improvements

Retrospectively, our dataset was not the most appropriate for image recolouring. Indeed, this dataset is not very colourful as many cats have colours that are close to grey.

The idea of restricting the task to a specific class is still interesting to us. However, a more appropriate dataset with more vibrant colours that are highly related to textures and shapes such as landscapes, fruits, or flowers would have given better results.

Moreover, we think that the amount of data we used for training was insufficient to obtain good generalisation. Therefore, it would have been interesting to train the model with a larger dataset. The number of epochs during training could be reduced to keep reasonable computing times. Indeed, 300 epochs were needed to train our model but with a larger dataset, this number could have been inferior.


[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2020.

[2] Menghua Zheng, Jiayu Xu, Yinjie Shen, Chunwei Tian, Jian Li, Lunke Fei, Ming Zong, and Xiaoyang Liu. Attention-based CNNs for image classification: A survey. *Journal of Physics: Conference Series*, 2171(1):012068, jan 2022.

[3] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization.

[4] An object can have several plausible colourisations, for

example, an apple could be green or red.

[5] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. pages 567–575, 2015.

[6] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. *European Conference on Computer Vision*, 2008, 09 2008.

[7] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. 2016.

[8] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.*, 35(4), jul 2016.

[9] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image colorization using generative adversarial networks. pages 85–94, 2018.

[10] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. Colorization transformer, 2021.

[11] https://www.kaggle.com/c/dogs-vs-cats.