Pro-CCaps: Progressively Teaching Colourisation to Capsules

Rita Pucci Christian Micheloni Gian Luca Foresti Niki Martinel
Università degli Studi di Udine, Italia
{rita.pucci, christian.micheloni, gianluca.foresti, niki.martinel}@uniud.it

Abstract

Automatic image colourisation studies how to colourise greyscale images. Existing approaches exploit convolutional layers that extract image-level features learning the colourisation on the entire image, but miss entities-level ones due to pooling strategies. We believe that entity-level features are of paramount importance to deal with the intrinsic multimodality of the problem (i.e., the same object can have different colours, and the same colour can have different properties). Models based on capsule layers aim to identify entity-level features in the image from different points of view, but they do not keep track of global features.

Our network architecture integrates entity-level features into the image-level features to generate a plausible image colourisation. We observed that results obtained with direct integration of such two representations are largely dominated by the image-level features, thus resulting in unsaturated colours for the entities. To limit such an issue, we propose a gradual growth of the reconstruction phase of the model while training. By advantaging of prior knowledge from each growing step, we obtain a stable collaboration between image-level and entity-level features that ultimately generates stable and vibrant colourisations. Experimental results on three benchmark datasets, and a user study, demonstrate that our approach has competitive performance with respect to the state-of-the-art and provides more consistent colourisation.

1. Introduction

In the colourisation problem, the input of a colouring algorithm is an image where the chrominance channels are absent. Only the luminance channel provides the initial information for colour reconstruction. Without the chrominance channels, the problem is ill-posed because of the multimodality of colourisation. This is also compounded by changes in illumination, variations in viewpoints, and occlusions. To reduce these issues, the understanding of the scene may be helpful, i.e., the grass is usually green, clouds are usually white, and the sky is blue. However, learning the scene as a whole might not be sufficient to properly model the appearance variations of entities, such as t-shirts, desks, and many other observable items can appear in slightly different shapes and colours. Existing works dealt with these problems by modelling the global content (image-level features) and objects' instances (entity-level features). Some works [3, 9, 25, 13, 4, 6, 10, 8] separately considered these representations that do not collaborate towards the final goal. The resulting entities' colourisations have unsaturated colours and smudges on boundaries. Others [17, 15, 22] considered both types of features that are first independently extracted, then merged in a different (later) stage. This neglects the importance of the interaction between the global content and the objects' instances in an image for its colourisation. *Differently*, we propose a novel model that encourages the collaboration between the entities and the global content of the image increasing the naturalness of generated colours. This is obtained by learning to fuse entity-level features, by capsule layers [21], with image-level features, by convolutional layers. The collaboration among features is implemented by residual and skip connections across features extraction (downsampling) and features reconstruction (upsampling) layers. Initial analysis show that the architecture obtains greater definition in colouring the entities and a greater naturalness of the image compared to the state of the art, although there is a strong imbalance in the generation of the colours. The colours for the entities looks unsaturated compared to those at the image-level (e.g. the blue of the sky looks more vibrant

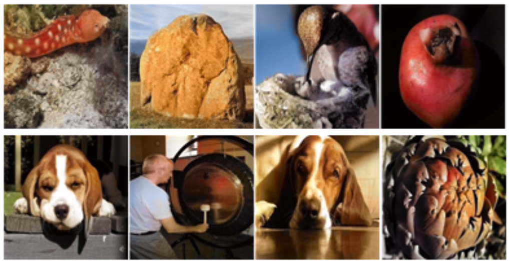

Figure 1. Colourisation results obtained with `Pro-CCaps`. Images have vivid and natural colours that are correctly located within different entities.

than the colours for objects' entities). As observed in [5], deep neural networks struggles in managing the high number of parameters in the reconstruction phase which leads to a prevalence of the reconstruction of image-level colours compared to those for the entities. To address such an issue, we propose to gradually increase model complexity for the reconstruction phase building on the top of the Progressive Growing while Learning scheme (ProGL) [11]. This training procedure lets the model learn the parameters at each step of reconstruction phase progressively by growing the complexity of the architectural structure over the course of epochs. This enabled us to obtain a better collaboration between image-level and entity-level features that yields to superior performance than the same structure trained in an end-to-end fashion.

To summarise, our `Pro-CCaps` model introduces the following contribution: (A) *a network architecture* that: (i) fuses the entity-level features captured through capsules with image-level features extracted from a hierarchical structure of convolutional layers; (ii) has a strong collaboration among layers by the means of skip and residual connections. (B) *a progressive learning procedure* that: (i) provides a prior knowledge based on the previous steps of progression; (ii) progressively integrate entity-level features reconstruction into image-level features ones; (iii) achieves vivid colourisation for each entity in the image.

We evaluated our model on three large scale benchmark datasets, namely ImageNet10k [20], COCOStuff [2], and Places205 [29]. Results demonstrate that our approach outperforms existing works in terms of image-quality metrics on ImageNet10k (e.g., see Fig. 1), while it is competitive on the other datasets. Through a large scale user study, we proved that the proposed method performs better than the closest approaches that use capsules [17] or two-stages interactions between image-level and entity-level features [22].

2. Related works

2.1. Colours reconstruction

Different algorithms were proposed for colourisation tasks. In [10], image colourisation is obtained by applying a general image-to-image translation with conditional adversarial networks. Other works [16, 23] propose generative models that in collaboration with semantic and prior knowledge predict a plausible colourisation. In [25, 13, 28, 27], the approaches focus on multi-task models and single-pixel significance to predict a right colourisation for the image. All these works focus on image-level features extraction, ending with attenuated colourisation for entities. A first attempt to address the entity-level features is introduced by the use of semantic labels together with image-level features to improve colourisation, as presented in [9, 28]. Conversely we apply unsupervised learning to train the proposed `Pro-CCaps`. Following the importance of semantic information and entity-level features, work at [25] introduces interpretable semantic by cross-channel encoding scheme, which is enriched in [15] by a pre-trained classification model. Works at [22, 17] are the closest to our proposed algorithm. In [22], the entity-level features is extracted by a dedicated model depict to identify bounding boxes of entities, while a parallel convolutional model extracts the image-level ones. The combination of the two distinct models requires a high number of parameter to be trained. In contrast, we propose a single model where the entity-level features extraction is integrated in one architecture. The work proposed in [17] applies the capsules layers for colourisation with no collaboration with convolutional layer. The model obtains promising results, but it is poor in definition. In contrast we propose a collaborative architecture trained in progressive learning that exploits outperforming results.

2.2. Progressive growing in learning

ProGL is a training methodology proposed by [11] for generative networks where it starts with low-resolution images, and then progressively increase the resolution by adding layers to the networks. This incremental nature allows the training to first discover the large-scale structure of the image distribution and then shift attention to increasingly finer scale detail, instead of having to learn all scales simultaneously. The idea of the growing model is based on the observation that the complex mapping from latent space to high-resolution images is easier to learn in steps. Hence, the growth of the model in the learning phase provides a progressive increase in the resolution of the generated images. Works like [11, 12] propose ProGL applied for GANs to generate images of humans. In [5], the ProGL is introduced to obtain a multi tasking method based on salient object detection. In our work the concept of growing is applied to let the model learn gradually the collaboration among features extracted at the different levels to reconstruct the colour information. To the best of our knowledge this is the first application of this procedure for the colourisation task.

3. Proposed method

As shown in Fig. 2, `Pro-CCaps` is composed of a stack of feature extraction blocks, i.e., ***the downsampling phase***, that consists of a *preprocessing block* (`PreB`), four *double block down* (`DBD`), and a *primary capsule down* (`PCD`) layer. This is followed by a stack of feature reconstruction blocks, i.e., the ***upsampling phase***, that consists of a *primary capsules up* (`PCU`), four *double block up* (`DBU`), and a *postprocessing block* (`PostB`). The structure ends with a block dedicated to ***channels reconstruction***, that consists of a *channels reconstruction block* (`CRB`). To enforce collabo-

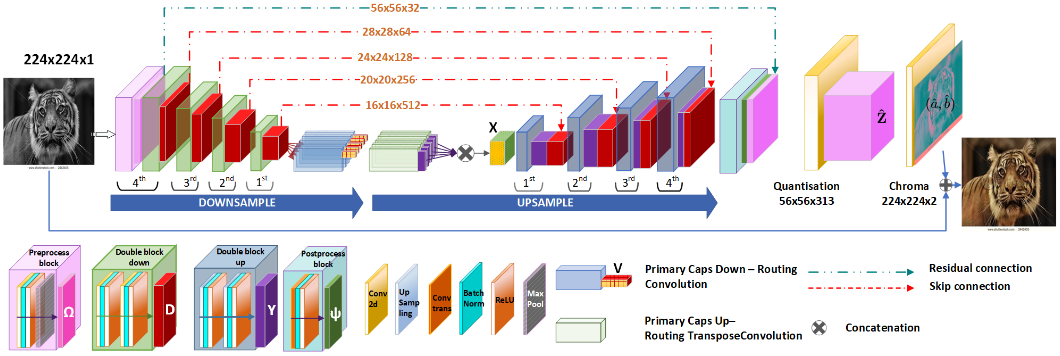

Figure 2. Proposed Pro−CCaps architecture. The downsample layers extract the image-level and entities-level features from the greyscale image. The upsample layers reconstruct the colours taking into consideration the features from the previous layer and from the corresponding layer in the downsample. The upsample structure is mirrored to the downsampling phase.

rations among layers in the two phases, we introduce skip connections between them. We also add a residual connection between the PreB and the PostB, to restore spatial information. An image $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ is represented in the CIELab colour space: $\mathbf{I}_L$ for lightness (input), and $\mathbf{I}_a$ and $\mathbf{I}_b$ for chrominance (generated).

3.1. Pro−CCaps Architecture

The downsampling phase To extract the image-level features from $\mathbf{I}_L$, we apply the combination of the stack *PreB*, and DBD^n, with $n \in [4, .., 1]$. Let $\mathbf{D}^n$ be the image-level feature representation generated by DBD^n. The five functional blocks form a 28-layers network emitting:

$$\mathbf{D}^1 = f_{\texttt{DBD}^1}(f_{\texttt{DBD}^2}(f_{\texttt{DBD}^3}(f_{\texttt{DBD}^4}(f_{\texttt{PreB}}(\mathbf{I}_L))))) \quad (1)$$

where $f_{\texttt{DBD}^n}$ denote the function of the DBD at layer n. The third functional block is the PCD, which consists of two layers of capsules. The first layer of capsules computes the activation vectors, it is formulated as:

$$\mathbf{U} = [\texttt{Flatten}(\texttt{Conv}_1(\mathbf{D}^1))^T, \cdots, \\ \texttt{Flatten}(\texttt{Conv}_C(\mathbf{D}^1))^T] \quad (2)$$

where C is the number of capsules and each column of $\mathbf{U}$ is the capsule output $\mathbf{u}_i \in \mathbb{R}^k$. The second layer of capsules applies an affine transformation on $\mathbf{U}$ with a weight matrix $\mathbf{W}_{ij} \in \mathbb{R}^{k \times \hat{k}}$, to obtain the prediction vectors at entities-level.

$$\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i \quad (3)$$

The prediction vectors are later grouped through the "routing by agreement" mechanism [21]. We perform three iterations, each of which is finalised to group by agreement the $\hat{\mathbf{u}}_{j|i}$ training the coupling coefficients $\mathbf{c}_{i|j}$ to identify clusters of features. The cluster of features identify entities, and these clusters are formally defined by the weighted sum of $\hat{\mathbf{u}}_{j|i}$ vectors:

$$\mathbf{v}_j = squash(\sum \mathbf{c}_{i|j} * \hat{\mathbf{u}}_{j|i}) \quad (4)$$

where $\mathbf{v}_j \in \mathbb{R}^{\hat{k}}$. The final output of downsampling is the matrix $\mathbf{V} = \mathbf{v}_0, \cdots, \mathbf{v}_j$, it carries information about how strong the capsules agree on the presence of an entity, we consider $\mathbf{V}$ the entities-level features.

Details of stacked layers in downsampling: the PreB is composed of a Conv−BN−ReLU−MaxPool. It achieves a reduction of resolution by a factor of two, from $\mathbf{I}_L \in \mathbb{R}^{224 \times 224 \times 1}$ to $\mathbf{\Omega} \in \mathbb{R}^{56 \times 56 \times 32}$ pixels where $\mathbf{\Omega} = f_{\texttt{PreB}}(\mathbf{I}_L)$. Each DBDis composed of two consecutive sequences of Conv−BN−ReLU, following [7] schema, where the Conv is a 3×3 heterogeneous convolution. The BN is a batch normalisation followed by the ReLU used to non-linearly transform the fused features, which is then used as the input of the following block. The last stacked DBD^1 outputs an image-level feature matrix of $\mathbf{D}^1 \in \mathbb{R}^{16 \times 16 \times 512}$. This output matrix is given to PCD that, through 32 capsules, generates $\mathbf{V} \in \mathbb{R}^{32 \times 8 \times 8 \times 128}$ feature matrix. We obtain 32 matrices, one each capsule, in these matrices are described the entity-level features matrix.

The upsampling phase Let $\mathbf{V}$ denote the entity-level features extracted by PCD that lack of information about their spatial displacement within the input datum. The spatial information is fundamental to localise the extracted features, hence to obtain a correct colourisation of the overall image. The PCU introduces a mechanism that "inverts" the PCD procedure to reconstruct the spatial information. We apply a weight matrix $\mathbf{W}^r_{ji} \in \mathbb{R}^{\hat{k} \times k}$, to obtain the reversed affine transformation:

$$\mathbf{u}^r_{i|j} = \mathbf{W}^r_{ji}\mathbf{v}_j \quad (5)$$

then $\mathbf{u}^r_{i|j} \in \mathbb{R}^k$ are stacked in $\mathbf{U}^r$ having the same size of $\mathbf{U}$. The $\mathbf{u}^r_i$ are the input of the ith reversed capsules, implemented with a transpose convolutional layer, (TConv$_i$) of the first layer that compose the PCU, it is formulated as:

$$\mathbf{X} = [\texttt{Reshape}(\texttt{TConv}_1(\mathbf{u_1^r})), \cdots, \\ \texttt{Reshape}(\texttt{TConv}_k(\mathbf{u_k^r}))] \quad (6)$$

where $\mathbf{u_i^r}$ denote the i^{th} row of $\mathbf{U}^r$. Matrix $\mathbf{X}$ consists of the initial colours reconstruction from entity-level features $\mathbf{V}$. $\mathbf{X}$ is the input to the stack layers $\texttt{DBU}^m$ where $m \in [1, \cdots, 4]$ and $\mathbf{Y}^m$ is the output of $\texttt{DBU}^m$. The $\texttt{DBU}^1$ receives in input only the $\mathbf{X}$ and it provides $\mathbf{Y}^1$ as formulated:

$$\mathbf{Y}^1 = f_{\texttt{DBU}^1}(\mathbf{X}) \quad (7)$$

We apply skip connection to promote the collaboration between the two phases, for the following $\texttt{DBU}^m$ where $m \in [2, \cdots, 4]$, the reconstruction is formulated as:

$$\mathbf{Y}^m = f_{\texttt{DBU}^m}(cat(\mathbf{Y}^{m-1}, \mathbf{D}^{m-1})) \quad (8)$$

The last upsampling layer (PostB) computes

$$\mathbf{\Psi} = f_{\texttt{PostB}}(cat(\mathbf{Y}^4, \mathbf{D}^4)) \quad (9)$$

where PostB performs the reversed function of PreB to obtain $\mathbf{\Psi} \in \mathbf{R}^{H \times W \times \Gamma}$ where dimensions corresponds to the dimensions of $\mathbf{\Omega}$. It consists of the final composition of all the features extracted in the UpSample phase. The five functional blocks consists of 28-layers network.

Details of stacked layers in upsampling: each DBU in the stack consists of UpSample followed by two consecutive sequences of Conv−BN−ReLU, the UpSample upsample the resolution of the input to match with the feature matrices at downsampling phase. The PostB is composed by TConv-BN-ReLU, which performs the opposite function of the PreB. We apply the five functional blocks, $\texttt{DBU}^m$ and PostB, from the input X, they output a feature reconstruction matrix of $\mathbf{\Psi} \in \mathbb{R}^{56 \times 56 \times 32}$.

Channels reconstruction The last block of the model is CRB. It reconstructs the colours in two different representation of the (a,b) channels of CIELab space. A quantised representation, based on the idea in [25], that prevents the solution be out of the set of gamut colours, giving implausible results, and the chroma representation lets the model predict consistent representation of colours in the a and b channels. Without the quantisation representation, the colours tends to obtain averaging effect in favour of greyish and desaturated results. On the other hand, without the chroma one, the model is unable to predict the representation of colours over the two channels. The CRB consists of stacked Q and Chroma layers to predict the colours of $\mathbf{I}_L$ in the two different spaces. To design Q, we followed [25] and quantised the in-gamut CIELab colours with $bin = 10$ to obtain 313 colour classes. This makes the task a classification problem: for each point in the input matrix, Q predicts a colour class. Q receives the the residual of $\mathbf{\Psi}$ and $\mathbf{\Omega}$ and maps it over the quantised colour distribution

$$\hat{\mathbf{Z}} = f_{Quantisation}(sum(\mathbf{\Psi}, \mathbf{\Omega})) \quad (10)$$

where $\hat{\mathbf{Z}} \in \mathbb{R}^{56 \times 56 \times 313}$.

The Chroma layer consists of a 1 × 1-Conv layer followed by bilinear upsampling to resize $\hat{\mathbf{Z}}$ by a factor of 4, hence to map $\hat{\mathbf{Z}}$ onto the two chrominance channels $(\hat{a}, \hat{b}) \in \mathbb{R}^{224 \times 224 \times 2}$.

3.2. Loss Function

Quantised Colours Loss Pro-CCaps learns a distribution over per-pixel colours. Towards such an objective, the matrix $\hat{\mathbf{Z}}$ is compared with the projection of ground-truth chroma channels on the quantised representation. $\mathbf{I}_{a,b}$ is converted by soft-encoding scheme in the quantised representation $\mathbf{Z}$. This is used to compute

$$\mathcal{L}_q = -\sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_q \mathbf{Z}_{h,w,q} log(\hat{\mathbf{Z}}_{h,w,q}) \quad (11)$$

where $v(\cdot)$ re-weights the loss for each pixel based on pixel colour rarity. We have considered the soft-encoding and the $v(\cdot)$ values introduced by [25].

Chrominance Loss We compute the chrominance error by minimising the difference between the real ($\mathbf{I}_{a,b}$) and the predicted ($\mathbf{I}_{\hat{a},\hat{b}}$) colour channels as:

$$\mathcal{L}_c = ||\hat{a} - a||_2^2 + ||\hat{b} - b||_2^2. \quad (12)$$

Combined Loss Our model optimises $\mathcal{L} = \mathcal{L}_q + \mathcal{L}_c$.

3.3. Progressive Growing While Learning

We introduce the ProGL concepts [11] on the upsampling phase of Pro-CCaps while training. The procedure is shown in Fig. 3. Every ρ epochs of training, we add one $\texttt{DBU}^m$, and a relative temporary channels reconstruction block ($\texttt{TCRB}^m$), and we remove the $\texttt{TCRB}^{m-1}$. Each $\texttt{TCRB}^m$ follows the structure proposed for CRB , where the resolution of $\hat{\mathbf{Z}}$ and $(\hat{a}, \hat{b})$ is equal to $\mathbf{Y}^m$, defined by the level of progression, Tab. 1. At the beginning of training, the UpSample consists only of PCD. Let PCU being the first layer of reconstruction, $\texttt{TCRB}^p$ provides $\hat{\mathbf{Z}}^p$ and $(\hat{a}, \hat{b})^p$ as shown in Fig. 3. In following layers, for $\texttt{DBU}^m$, we add $\texttt{TCRB}^m$ and it provides $\hat{\mathbf{Z}}^m$ and $(\hat{a}, \hat{b})^m$. The last layer of growing is the PostB that complete the structure. We train the model with $\mathcal{L}$, described in Section 3.2.

Motivation: While the training progresses, the network retains a pool of pre-trained models [11], and learns lateral connections between the image and entity level features.

Table 1. Performance metrics on ImageNet10k computed for each step in the ProGL training procedure. The last four columns compare the final obtained by `Pro-CCaps`and `CCaps` (same architecture as `Pro-CCaps` end-to-end trained [19, 18]) with (*) and without fine-tuning on COCOStuff. We highlight in bright green the best result, in light green the second best, and in light blue the third best.

-	`PCU`	`DBU`^{1}	`DBU`^{2}	`DBU`^{3}	`DBU`^{4}	`Pro-CCaps` `PostB`	`Pro-CCaps`* `PostB`	`CCaps` -	`CCaps`* -
dim	15×15	16×16	20×20	24×24	28×28	224×224	224×224	224×224	224×224
PSNR↑	31.447	31.435	31.457	31.179	31.089	31.562	**32.910**	31.749	32.528
SSIM↑	0.972	0.973	0.968	0.955	0.953	0.960	**0.977**	0.974	0.975
LPIPS↓	0.065	0.066	0.066	0.078	0.081	0.083	**0.048**	0.055	0.051

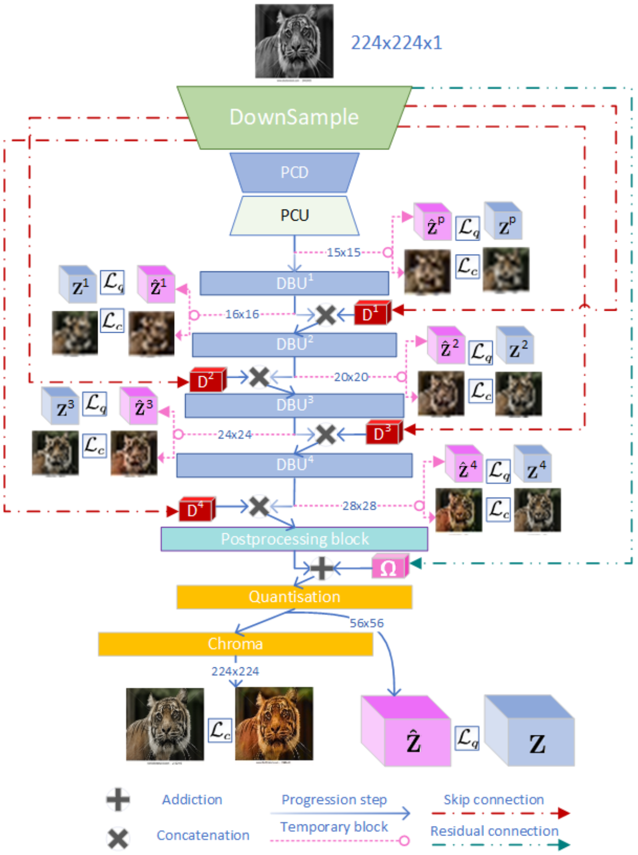

Figure 3. Overall architecture of the proposed progressive learning in upsampling phase. The model learns progresses gradually; at each step the model provides a quantised matrix $\hat{\mathbf{Z}}^\alpha$ where $\alpha \in \{p, 1, 2, 3, 4\}$ is the upsampling level and a proposed colourisation.

The ProGL proposes a solution to the insufficiency of the receptive field. In a non ProGL scheme, while the resolution increases, the `DBU`s are not able to learn the context of the entity. This is due to the limitations of the receptive field that tends to mix information. This difficulty affects mainly early layers vanishing the benefit of capsule layers. Moreover, the last layer has a high number of parameters to learn due to the high resolution of output, that makes difficult to learn and extract informative features that can be effective for the previous layers; the extracted information about colours at high resolution can be confusing for low resolution. Applying ProGL, the `Pro-CCaps`achieves richer compositionality of colours and allows prior knowledge to be integrated at each layer of the feature hierarchy. We think these properties let each `DBU` acquires the ability to reconstruct colours focusing onto a small number of parameters at the time, and reducing the confusion in colourisation. The `DBU`$^{(1,2)}$ build prior knowledge that is used by the successive layers to understand the colourisation of entities in the image. The `DBU`$^{(3,4)}$ focus on the context of the image, merging image-level features at high resolution with the prior knowledge, as observed in Fig. 4.

4. Experimental results

4.1. Datasets

Following SOTA methods [25, 22, 13], we considered the ImageNet dataset [20] to train our model on the 1.3M images (with no labels). The input images were resized to 224 × 224 and projected onto the CIELab colourspace. To assess model performance, we considered three benchmark datasets. ImageNet10k [13] is the test split of ImageNet dataset and it consists of 10k images, COCOStuff [2] contains a wide variety of natural scenes with multiple objects present in the 118k images. We use the training split for fine-tuning procedure and the provided validation split containing 5000 images for evaluation. Places205 [29] is a scene-centred dataset containing samples from 205 different categories. We considered the 20500 validation images for evaluation.

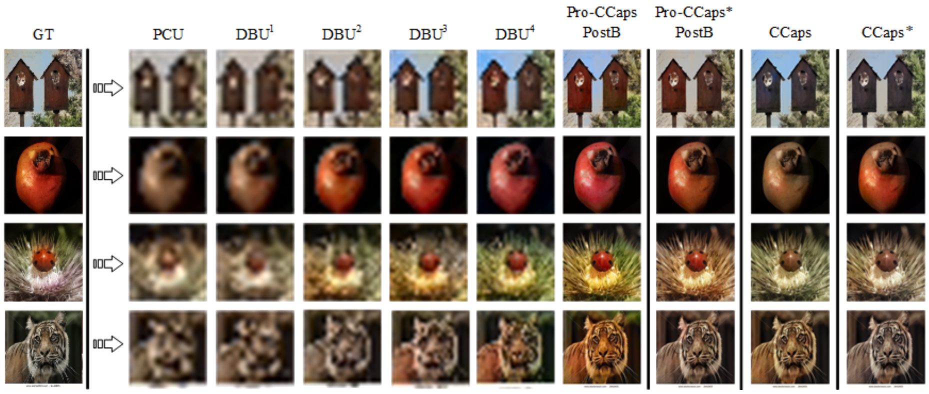

Figure 4. Reconstruction of colours in progressive learning. The GT column presents four samples from the ImageNet10k test set. The following five columns, show the results obtained at each step of the ProGL procedure. The `Pro-CCaps`, and `Pro-CCaps*` columns depict the final results. These are also compared with the results obtained by running our model without ProGL (`CCaps` and `CCaps*` columns). The ”*” identifies models fine-tuned on COCOStuff dataset

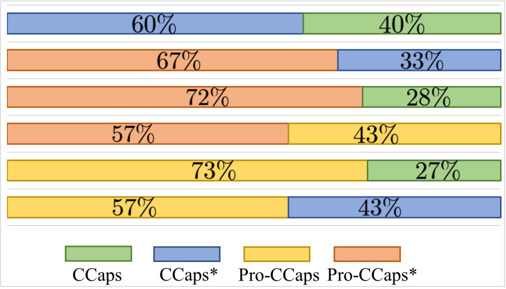

Figure 5. Summary of results obtained user study: we investigate the naturalness of images coloured by CCaps, or `Pro-CCaps`, and w/o fine tuning “ $*$ ”. The percentage reported in this figure represents the preference of users.

4.2. Implementation details[1]

In the training process, we used a batch of 32 samples and the Adam optimiser with a learning rate of $2e^{-3}$. We set $\rho = 10$, hence run 60 epochs of ProGL, then trained the whole network for additional 10 epochs.

Additional configuration: To better assess the benefits of our contributions, we perform additional experiments on the model for ablation study: let `Pro-CCaps*` and `CCaps` denote the model *fine-tuned* on COCOStuff, for comparison with [22], and the model trained in end-to-end scheme [19, 18], respectively. For the former, we fine-tuned the ImageNet pretrained `Pro-CCaps` for 35 epochs. End-to-end (no ProGL) training run for 40 epochs (experimentally proved to be optimal) with a batch size of 32. Both the fine tuning and the end-to-end apply the Adam optimiser with learning rate $2e^{-3}$.

4.3. Evaluation metrics

To assess our colourisation performance, we follow the experimental protocol in [14] and consider the Peak Signal to Noise Ratio (PSNR), the Learned Perceptual Image Patch Similarity (LPIPS) [26] (version 0.1 with VGG backbone), and the Structural Similarity Index Measure (SSIM) [24].

4.4. Ablation study

ProGL To investigate the benefits of ProGL, we conducted an analysis of the result obtained at each progression step. At each step, we expect to be able to predict the colourisation for the input image with the corresponding resolution, defined as *dim* in Table 1. Table 1 shows the achieved performance on ImageNet10k dataset. Results indicate an inverse relationship between the performance metrics and the depth of the network (i.e., from `PCU` to `PostB`). This is due to the spatial resolution of the generated colour image that increases with model depth. `PCU` predicts a 15×15 image, while `PostB` emits 224×224 pixels.The last four columns, show the metrics obtained by the complete model. The results indicates that `Pro-CCaps*` outperforms all the other models, while it seems to obtain competitive results in metrics when we do not apply fine-tuning on `Pro-CCaps`. To further understand the contribution of ProGL, present visual results shown in Fig. 4. In column `CCaps`, the colourisation obtained for the images is well defined for the global image, e.g. the background, but the entities colours are not saturated, almost brownish. This issue is shown in attenuated when with `CCaps*`. On the contrary, the colourisation obtained as result of the progression steps, column `Pro-CCaps-PostB`, is saturated and vivid for all the entities in the image. We observe that, in the first three steps of progression the colourisation focus on the entities (the entities are colourised while the background is brownish), in the last three steps the entities colourisation

[1] `https://github.com/Riretta/Pro_CCaps-Progressive-learning-with-capsules`

Table 2. Performance comparison with standard colourisation metrics. Models are trained on ImageNet. Last three rows ("*") report on the results with models fine-tuned on COCOStuff. Column "Param" reports the number of trainable parameters where known. In bright green the best result, in light green the second best, and in light blue the third best.

	ImageNet ctest10k			COCOStuff validation split			Places205 validation split			
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	Param
Pro-CCaps (our)	0.083	31.562	0.960	0.155	30.273	0.905	0.152	30.323	0.909	30 M
Larsson [13]	0.188	24.93	0.927	0.183	25.06	0.930	0.161	25.72	0.951	-
Iizuka [9]	0.200	23.63	0.917	0.185	23.86	0.917	0.146	25.58	0.950	-
Zhang [27]	0.145	26.166	0.932	0.138	26.823	0.937	0.149	25.823	0.948	34 M
Su [22]	0.134	26.98	0.933	0.125	27.77	0.940	0.130	27.16	0.954	103 M
Pro-CCaps* (our)	0.048	32.910	0.977	0.126	30.713	0.921	0.126	30.711	0.926	30 M
Zhang * [27]	0.140	26.482	0.932	0.128	27.251	0.938	0.153	25.720	0.947	34 M
Su * [22]	0.125	27.562	0.937	0.110	28.592	0.944	0.120	27.800	0.957	103 M

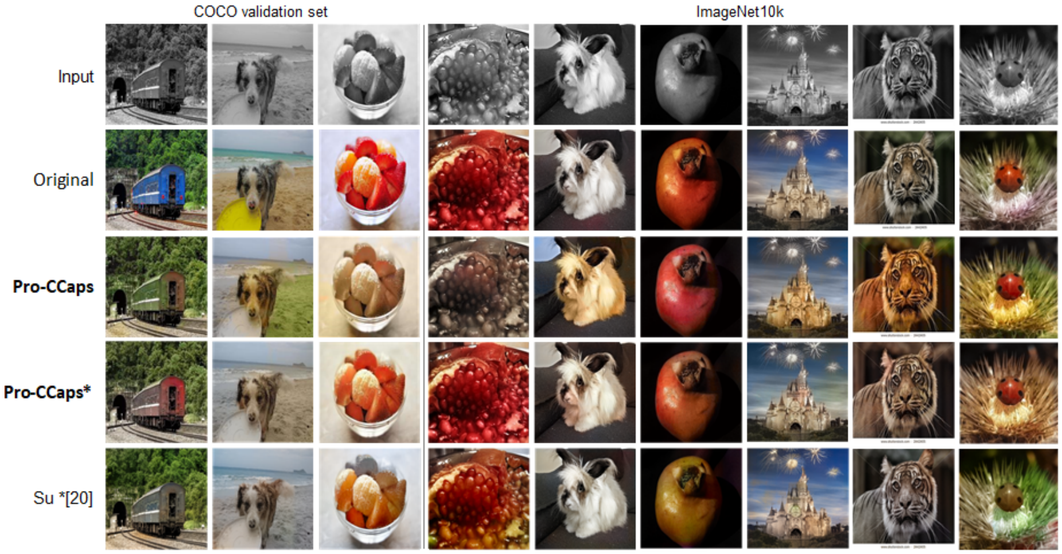

Figure 6. Qualitative comparisons among Pro-CCaps, Pro-CCaps*, and [22] on COCO validation set and ImageNet10k. Results are shown for best performing models, as shown in Tab. 2.

is well defined and has a high influence in the colourisation of the image. The colours generated by Pro-CCaps* are still coherent with entities in the images. There are also no smudges on object's edges. More interestingly, qualitative results by such two methods seems to be more realistic than the ones produced by CCaps (and CCaps*), thus posing some questions on the existing metrics. To gather more insights about such an outcome, we have conducted a large scale user-study.

User study The perceptual realism is how compelling the colours look to a human observer. Users were asked to choose the more appealing colourisation among the ones generated by our Pro-CCaps and CCaps models (w/ and w/o fine-tuning). We gathered 2640 answers from 132 users. Results shown in Fig. 5, demonstrate that images generated by Pro-CCaps and Pro-CCaps* are more appealing to the users a preference of 73% and 72% of the times compared to CCaps. All the metrics in Tab. 1 show that CCaps and CCaps* are performing better than Pro-CCaps . Thus, there are some discrepancy between the considered metrics and the users answers (despite some agreement on Pro-CCaps* results). Such outcomes substantiate our initial doubts on the existing performance metrics. Proposing a new metric is out of the scopes of this work, but we believe that community efforts are needed to

Table 3. Metrics results on DIV2K dataset [1]: we compare Pro-CCaps, and Pro-CCaps*with the two latest and closest works at the SOTA - [17, 22].

	Pro-CCaps*	Pro-CCaps	[22]	[17]
PSNR↑	30.40	30.39	30.04	21.08
SSIM↑	0.91	0.91	0.88	0.85

propose more systematic ways of assessing colourisation performances.

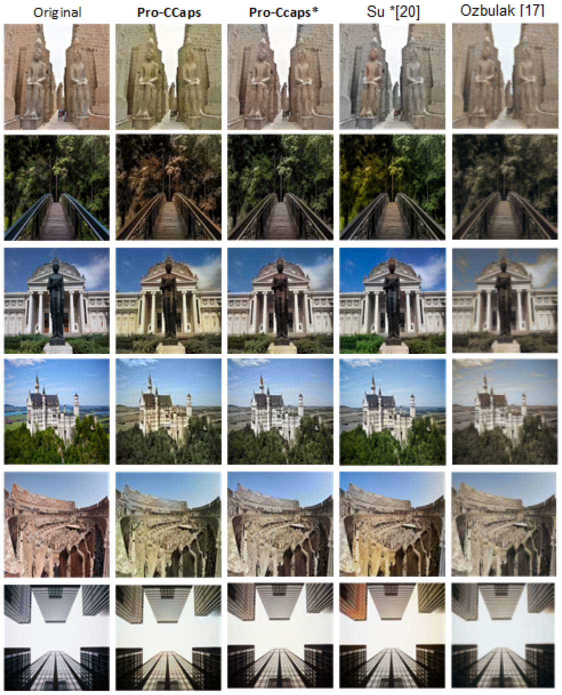

Figure 7. Qualitative comparison of Pro-CCaps with DIV2K dataset [1]. Pro-CCaps, and Pro-CCaps* are compare with [17] and [22] considered the closest for intent and structures to our proposed work.

4.5. State-of-the-Art Comparisons

In the following we compare the performance of our approach with SOTA methods [13, 9, 27, 17, 22]. In particular, we focus the comparison on [22, 17] that are the closest methods to our work. In [22], a pre-trained colourisation model [25] is combined with a (COCO pre-trained) object detector to focus on entities. [17] is the first paper that exploited capsule networks for colourisation. The model maintains the original structure of CapsNet [21] with a different generation head. In contrast to both works, we propose an approach that (requires no pre-training and) extracts and enforces collaboration between object entities and image features within a single architecture. This is also trained with ProGL to develop a strong collaboration among each part of the whole architecture. **Quantitative Comparisons:** Performance comparisons on the common benchmark colourisation metrics and datasets are presented in Tab. 2 and Tab. 3. As shown in Tab. 2, both Pro-CCaps and Pro-CCaps* outperform the SOTA on all the three metrics on the Imagenet10K dataset. Pro-CCaps and Pro-CCaps* achieve an LPIPS score of 0.083 and 0.048, respectively, thus improving by more than 4% the previous SOTA results obtained by [22]. Moreover, the PSNR obtained with both our models outperforms the SOTA with COCOstuff and Places205, where the models achieve a competitive performance on SSIM and LPIPS. It should be noted that our model uses a third of trainable parameters compared to [22], as shown in the last column of Table 2. In Tab. 3, we compare the performance of our method on the DIV2K dataset limited to PSNR and SSIM (to properly compare with [17]). Results show that both our proposed models outperform existing models, thus demonstrating that our approach performs very well on different datasets. **Qualitative Comparisons:** Fig. 6 and 7 compare the colourisation results generated by our method with existing SOTA works. In Fig. 6, we show the results obtained with the proposed models (Pro-CCaps and Pro-CCaps*) and compare with the best SOTA one [22]. Results demonstrate that colour images generated by Pro-CCaps and Pro-CCaps* are less blurry with more saturated and vibrant colours. Fig. 7 provides additional comparisons with the closest related works [17, 22]. For a fair comparison with [17], coloured images are generated for the DIV2K dataset. Results demonstrate that Pro-CCaps* provides a sharper and more consistent colourisation on object entities than [17, 22]. These substantiate the importance of collaboration between entity and image-level features learned through a ProGL scheme.

5. Conclusion

In this paper, we proposed a Progressive learning for Colourisation with Capsules (Pro-CCaps) for greyscale images. Pro-CCaps combines image-level and entities-level features by cascading several types of modular blocks to extract features at different levels and resolutions. The application of ProGL to Pro-CCaps prevents possible training instability and performance degradation caused by upsampling operations. We have proposed an architecture that promotes a features fusion scheme based on skip connections and residual connection to address the long-term dependency problem as well as prevent information loss so as to significantly improve the quantitative and qualitative reconstruction of colours. Comprehensive evaluations on three benchmark datasets demonstrate that our approach provides a good visual quality among different visible features in the images (e.g. landscape, portrait, macro).

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018.

[3] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[4] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David Forsyth. Learning diverse image colorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6837–6845, 2017.

[5] Joonki Paik Dong-Goo Kang, Sangwoo Park. Coarse to fine: Progressive and multi-task learning for salient object detection. In *2020 - International conference of pattern recognition*, 2020.

[6] Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy. Pixcolor: Pixel recursive colorization. *arXiv preprint arXiv:1705.07208*, 2017.

[7] Isma Hadji and Richard P Wildes. What do we understand about convolutional networks? *arXiv preprint arXiv:1803.08834*, 2018.

[8] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):1–16, 2018.

[9] 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 Transactions on Graphics (ToG)*, 35(4):1–11, 2016.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[12] Natsumi Kato, Hiroyuki Osone, Kotaro Oomori, Chun Wei Ooi, and Yoichi Ochiai. Gans-based clothes design: Pattern maker is all you need to design clothing. In *Proceedings of the 10th Augmented Human International Conference 2019*, 2019.

[13] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016.

[14] Chenyang Lei and Qifeng Chen. Fully automatic video colorization with self-regularization and diversity. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] Thomas Mouzon, Fabien Pierre, and Marie-Odile Berger. Joint cnn and variational model for fully-automatic image colorization. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 535–546. Springer, 2019.

[16] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image colorization using generative adversarial networks. In *International conference on articulated motion and deformable objects*, pages 85–94. Springer, 2018.

[17] Gokhan Ozbulak. Image colorization by capsule networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[18] Rita Pucci, Christian Micheloni, Gian Luca Foresti, and Niki Martinel. Is it a plausible colour? ucapsnet for image colourisation. In *Workshop: Self-Supervised Learning – Theory and Practice, NeurIPS 2020*, 2020.

[19] Rita Pucci, Christian Micheloni, and Niki Martinel. Collaborative image and object level features for image colourisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2160–2169, 2021.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[21] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[22] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] Patricia Vitoria, Lara Raad, and Coloma Ballester. Chromagan: Adversarial picture colorization with semantic class distribution. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2445–2454, 2020.

[24] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[25] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[26] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[27] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

[28] Jiaojiao Zhao, Li Liu, Cees GM Snoek, Jungong Han, and Ling Shao. Pixel-level semantics guided image colorization. *arXiv preprint arXiv:1808.01597*, 2018.

[29] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.