Collaborative Image and Object Level Features for Image Colourisation

Rita Pucci
University of Udine
Udine, Italy
rita.pucci@uniud.it

Christian Micheloni
University of Udine
Udine, Italy
christian.micheloni@uniud.it

Niki Martinel
University of Udine
Udine, Italy
niki.martinel@uniud.it

Figure 1: Achieved automatic colourisation results. We introduce a colourisation model that promote the collaboration between image (context) and object features to produce plausible and colourful results on different scenes with no human intervention.

Abstract

Image colourisation is an ill-posed problem, with multiple correct solutions which depend on the context and object instances present in the input datum. Previous approaches attacked the problem either by requiring intense user-interactions or by exploiting the ability of convolutional neural networks (CNNs) in learning image-level (context) features. However, obtaining human hints is not always feasible and CNNs alone are not able to learn entity-level semantics, unless multiple models pre-trained with supervision are considered. In this work, we propose a single network, named UCapsNet, that takes into consideration the image-level features obtained through convolutions and entity-level features captured by means of capsules. Then, by skip connections over different layers, we enforce collaboration between such the convolutional and entity factors to produce a high-quality and plausible image colourisation. We pose the problem as a classification task that can be addressed by a fully unsupervised approach, thus requires no human effort. Experimental results on three benchmark datasets show that our approach outperforms existing methods on standard quality metrics and achieves state-of-the-art performances on image colourisation. A large scale user study shows that our method is preferred over existing solutions. Code available at https://github.com/Riretta/Image_Colourisation_WiCV_2021.

1. Introduction

Adding a plausible colourisation to a monochromatic (greyscale) image through an autonomous system is a challenging research topic in computer vision and pattern recognition [36]. The colourisation task has multiple plausible choices due to the multi-modality of appearance of entities (e.g., a car can be red, blue, green) [3]. Automatic colourisation has significant impacts on historical image/video restoration and image compression [22], while also being a useful proxy task for learning visual representations [26]. Previous works relied on intense user-interaction procedures to guide the colourisation through scribbles-based [21, 14, 36, 14, 23, 40, 29] and image references-based [11, 41, 13, 7] methods. Fully automatic solutions were proposed by introducing different deep architectures [33, 6, 15, 4, 10, 13, 17, 37, 19, 27, 24] which either considered image-level or entity-level features alone, thus neglecting the importance of the interaction between the global content and the object instances in an image. Such methods also heavily relied on pre-trained models and supervised learning tasks to learn object semantics.

On the contrary, we hypothesise that the colourisation process can be tackled through (i) a single model that (ii) captures image context and the object entities (iii) without supervision to generate a plausible colourisation[1]. *These considerations motivate us to introduce a novel approach*

[1] i.e., that has geometric, perceptual, and semantic photo-realism.

that performs automatic colourisation by combining the image context with the object instances, without the need of labelled data. To capture such factors, we leverage convolutional and capsules layers [32]. The former allows us to identify and extract features which carry information about the image context. Through the routing by agreement routine, the latter captures the presence and the features of object entities. Skip connections over different layers are used to encourage a collaboration between convolutional representations, thus allowing us to leverage image-level and entity-level features to produce a plausible colourisation. We considered the fact that any colour photo can be used as a training sample by taking the image's lightness channel as input and its colour channels as the supervisory signal. Following such an intuition, we pose the colourisation problem as a colourisation task that learns a distribution of colours for each pixel. By exploiting such a distribution through a trainable mapping function, we capture the fact that a same object can have different colourisations (e.g., an apple can be red, green or yellow).

Concretely, our contribution is a novel fully automatic image colourisation approach that (i) learns image-level and entity-level representations to grasp information about the image context and object entities; (ii) exploits skip connection at different network depths to enforce collaboration between the features of downsampling and upsampling phases; (iii) captures the multi-modality of the colourisation problem by learning to predict a distribution of plausible colours; (iv) sets the learning problem to tackle the colourisation problem with neither human intervention nor pre-trained models.

We have conducted a comprehensive evaluation of our model on three large scale benchmarks datasets, namely ImageNet [31], COCOStuff [2] and Places205 [42]. We also assessed the qualitative performance of the proposed approach through a large scale human-based evaluation study. Results demonstrate that our approach outperforms existing works in terms of the image colourisation quality both using common image-quality metrics as well as with respect to the human preference. We also show its advantages in generating plausible colourisation of different entities over works leveraging labelled data to learn object semantics.

2. Related works

Existing image colourisation works are either based on scribble and example images, or rely on learning processes.

Based on Scribbles and Example Images. The multi-modal nature of image colourisation problem, was initially tackled with local hints provided by humans [21, 14, 29, 23, 36, 40]. These approaches propagate similar colourisations over specific areas, recognised with low-level similarity metrics. In [21], pixels with similar luminescence should have a similar colour, given by the scribble. This method suffers of colours overcoming object's edges, later limited by means of edge detection [14] and texture similarity [29, 23]. An up-grade was proposed in [36] by relaxing constraints on the position of scribbles. In [40], scribbles and human hints were jointly exploited to limit the manual effort. Other works explore the "example image" technique where colours are transferred from a reference (example) image to a greyscale image. The reference images are specified by users or searched on internet [35, 3, 11, 5, 16, 13]. Information is transferred between reference and input images on the basis of low-level similarity metrics measured at pixel level [35], semantic segments level [3] or super-pixel level [11, 5]. A combination of scribbles and example images was proposed in [16, 13].

Even if the achieved results are interesting for some specific tasks, the required human effort is too intensive and generally not affordable when the colourisation objective considers large scale problems. Our approach is fully automated, thus suitable for large scale colourisation tasks.

Based on Learning Processes. The colourisation task through machine learning has received great attention in the recent past [15, 41, 37, 19, 24, 17, 34, 7, 6, 4, 27, 10]. In [15, 41] semantic labels are exploited to capture entity-level features that, together with image-level features, improve the colourisation. In [37], semantic interpretability is obtained by a cross-channel encoding scheme, later exploited in [24] with a pre-trained classification model. Generative models [17, 25, 34] were also exploited together with pre-trained classifiers. Other approaches [37, 19, 41] focused on multi-task models and single pixel significance. Recently, more focus was devoted to capture entity-level features by embracing the capsules concept [27] or by exploiting pre-trained object detection models [33]

Such approaches either separately considered image-level and entity-level features or hinge on models pre-trained through supervised approaches. In contrast, we jointly capture the two representations and introduce a model that is trained with chromatic channels only.

The works in [27] and [33] are the closest to our approach. Differently from these, we introduce a single architecture that (i) captures entity-level features through capsules without using any prior knowledge to detect and extract object information; (ii) obtains the spatial information discarded by capsules through image-level features from CNNs; (iii) tackles the emerging problem of figure-ground separation by enforcing collaboration between the image-level and entity-level features with the introduction of skip connections among layers computing such two representations. We also introduce (iv) a combination of a colour quantisation loss [37] to learn a distribution of colours and a colour error loss that allows us to deal with multiple plausible colourisations of a same/similar object.

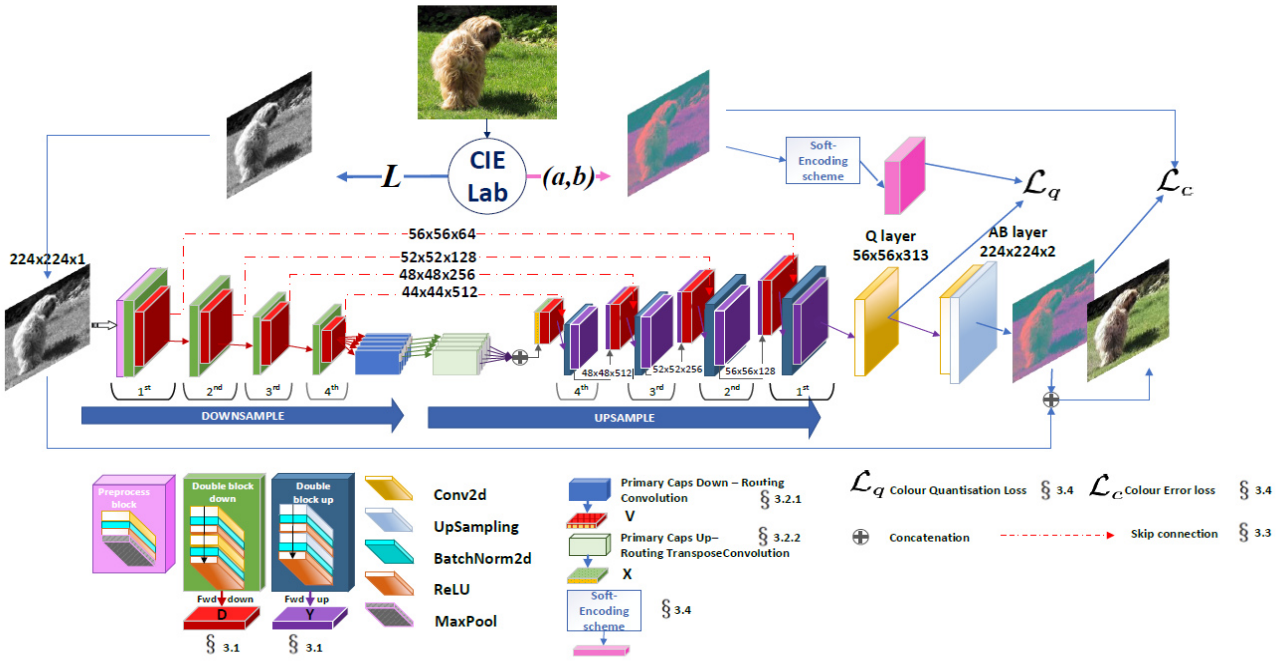

Figure 2: Our proposed UCapsNet architecture takes the L channel of a CIELab image and processes it by a set of *double block down* layers detecting image-level features. Then *primary caps down* layers extracts the entity-level features. Such representations at image and object levels are then considered in a quantisation layer that allows us to learn a colour distribution, hence to generate plausible (a, b) colours channels.

3. Proposed Approach

Our goal is to generate a plausible colourisation for a grayscale input image. Towards such a goal, our architecture, shown in Figure 2, starting from the CIELab lightness channel $L \in \mathbb{R}^{H\times W\times 1}$ learns to predict the corresponding colour channels $(a, b) \in \mathbb{R}^{H\times W\times 2}$. These represent the chrominance channels signal used in training procedure.

The downsample phase is responsible of learning image-level and entity-level representations. The former is obtained through a preprocessing block followed by a set of consecutive *double block down* operators which include convolutional layers. The latter is computed by the *primary capsule down* operator taking the image-level features to encode the entity-level ones into capsules.

The upsample phase leverages the image-level and entity-level representations to generate a plausible image colourisation. The *primary capsules up* operator decodes the input capsules to transform them into spatial features carrying information about object entities. Such features are combined with image-level features obtained through skip connections, then processed by a set of consecutive *double block up* operators. The model learns a colour distribution over pixels ($\mathbf{Q}$ layer) that is later exploited to predict the colour channels ($\mathbf{AB}$ layer).

3.1. Image-Level Features

Convolutional layers extract a feature map indicating the location and strength of a detected feature in an image. These define the image-level representation carrying the context information that is important to perform a suitable colourisation. We obtain such a representation through the introduction of a preprocessing block, followed by the *double block down* (`DBD`). The former is composed of a `Conv–BN–ReLU–MaxPool` sequence. To design the latter, we followed a common practice [12] by which image-level features can be obtained through a hierarchical structure of repeated convolutional, normalisation, and non-linearity layers. Thus, we let a `DBD` be composed of two consecutive sequences of `Conv -BN-ReLU` layers. Four `DBD` are stacked in the downsample phase with $\mathbf{D}_i$ being the output of $\texttt{DBD}_i$, for $i = 1, \cdots, 4$.

3.2. Entity-Level Features

While image-level features are important to capture information about the context of an image, the colourisation process also hinges on the entities present in the scene. To capture entity-level features, we exploit the concepts behind capsules [32], i.e., groups of neurons generating vectors that indicate the probability of existence of object entities.

Algorithm 1: Routing by Agreement [32]

Input: Weighted capsule features $\hat{\mathbf{u}}_{j|i}$
Output: Entity-level feature vector: $\mathbf{v}_j$
for all capsule i in the first layer and capsule j in the second layer: $b_{ij} \leftarrow 0$
for *routing iterations* **do**
  for each capsule i in the first layer: $\mathbf{c}_i \leftarrow \text{softmax}(\mathbf{b}_i)$;
  for each capsule j in the second layer: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$; $\mathbf{v}_j \leftarrow \text{squash}(\mathbf{s}_j)$; ▷squash as in [32]
  for each capsule i in the first layer and capsule j in second layer: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}\mathbf{v}_j$
end

3.2.1 Primary Caps Down (PCD)

The PCD operator is composed of two capsule layers. The first layer of capsules computes $\mathbf{U} = [\texttt{Flatten}(\texttt{Conv}_1(\mathbf{D}_4))^T, \cdots, \texttt{Flatten}(\texttt{Conv}_k(\mathbf{D}_4))^T]$. With each column of $\mathbf{U}$ being the capsule output $\mathbf{u}_i \in \mathbb{R}^k$. To identify entity-level features, a weight matrix $\mathbf{W}_{ij} \in \mathbb{R}^{k\times\hat{k}}$ is introduced to obtain the second layer capsules $\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i$, later grouped through the "routing by agreement" mechanism in Algorithm 1. At each iteration of the routine, $\hat{\mathbf{u}}_{j|i}$'s are grouped in agreement with the coupling coefficient $\mathbf{c}_i$ to identify clusters of features, i.e., the entity-level feature vector $\mathbf{v}_j \in \mathbb{R}^{\hat{k}}$. This carries information about how strong the capsules agree on the presence of an entity.

3.2.2 Primary Caps Up (PCU)

The PCD capsule outputs $\mathbf{v}_j$'s contain entity-level features but lack details about their spatial displacement. Since this information is fundamental for the colourisation task, we introduce a mechanism that inverts the PCD procedure to reconstruct the spatial information. We introduce a weight matrix $\mathbf{W}^r_{ji} \in \mathbb{R}^{\hat{k}\times k}$ connecting each PCD output to the PCU capsules. This computes $\mathbf{u}^r_i = \mathbf{W}^r_{ji}\mathbf{v}_j$, that are then stacked to obtain $\mathbf{U}^r$ –with the same size of $\mathbf{U}$. The resulting k rows of $\mathbf{U}^r$ are then reshaped, processed by k independent TransposeConv operators, then concatenated to obtain an output matrix $\mathbf{X}$ having the same dimensionality of $\mathbf{D}_4$. Thus, $\mathbf{X}$ contains spatial information generated through the entity-level features obtained through capsules.

3.3. Image-Level and Entity-Level Collaboration

Through the introduction of the capsule layers, we are able to extract entity-level features. However, generating the colourisation output requires a precise answer at pixel level, which is likely not to be directly achievable through the PCU layer alone (see § 4.3). Thus, to promote such an output, we enforce collaboration between image-level and entity-level features by introducing a mechanism to project the image-level features learnt at different stages of the downsample phase into the pixel space [30]. This is achieved by introducing skip connections from the downsample phase to the upsample one. The upsample phase is composed of four *double block up* (DBU) operators designed following the same considerations adopted for the DBD, hence composed of two consecutive sequences of UpSample-BN-ReLU layers. As shown in Figure 2, $\texttt{DBU}_i$, with $i = 4, \cdots, 1$ receives as input the concatenation of $\mathbf{D}_i$ and the output of the preceding $\texttt{DBU}_{i+1}$, denoted as $\mathbf{Y}_{i+1}$. $\texttt{DBU}_4$ processes the concatenation between the de-routed entity-features in $\mathbf{X}$ and $\mathbf{D}_4$. The up-sampling operations in each DBU allow us to reconstruct the (a, b) channels having the same size of the input image. The skip connections enforce exploitation of higher resolution features that can be missing due to the sparsity of the up-sampling operations.

3.4. Objective Function

A same/similar object can have different colourisations. To allow the model capture such a multi-modality of appearance of entities, we propose a **Q** layer to learn a distribution over quantised pixel colours. However, learning such a distribution is not the final objective of the colourisation process. We ultimately want the model to generate the chrominance (a, b) colours for the L input. Towards such a goal, we exploit the learned distribution to produce a plausible colourisation. This is achieved through the **AB** layer learning a mapping from the quantised space to the chrominance one by means of the colour error loss.

Colour Quantisation Loss. To generate a plausible colourisation, we want to learn a distribution over per-pixel colours. Towards such an objective, inspired by [37], we quantised the (a, b) space into bins with grid size 10. We then kept only the $Q = 313$ values which are in-gamut. These denote the distinct classes a pixel can belong to. Starting from the input channel L, our model learns to generate a distribution over such classes. This is achieved by introducing the **Q** layer composed of UpSample and 1×1-Conv layers, processing the $\mathbf{Y}_1$ feature map to predict the colour distribution $\hat{\mathbf{Z}} \in \mathbb{R}^{H\times W\times Q}$. This is used to compute the quantisation loss

$$\mathcal{L}_q = -\sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_q \mathbf{Z}_{h,w,q} log(\hat{\mathbf{Z}}_{h,w,q}) \qquad (1)$$

where $\mathbf{Z}_{h,w,q}$ is the ground-truth colour distribution for the (h, w) pixel obtained through a soft-encoding scheme and $v(\cdot)$ re-weights the loss for each pixel based on pixel colour rarity. We have considered the soft-encoding and the $v(\cdot)$ values introduced [37].

Table 1: Quantitative comparison. Top rows show the results achieved by approaches that do not consider labels for training nor models pre-trained with supervision (i.e., unsupervised/self-supervised methods). Bottom rows show the performance obtained by methods that need semantic labels for training or require models pre-trained with supervision (i.e., supervised methods). Columns follow a red-to-green colour-coded representation: the better the performance the greener the table cell. Best results for each of the two groups are in bold. Best overall results are also underlined. (*Best viewed in colours.*)

Method	ImageNet ctest10k		COCOStuff validation split		Places205 validation split		Publication
	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	
UCapsNet	**0.141**	**<u>30.42</u>**	**0.132**	**<u>30.59</u>**	**0.131**	**<u>30.61</u>**	Proposed
Lei *et al.*	0.202	24.52	0.191	24.59	0.175	25.07	CVPR2019 [20]
Isola *et al.*	–	21.57	–	–	–	–	CVPR2017 [17]
Zhang *et al.*	0.238	22.04	0.234	21.83	0.205	22.58	ECCV2016 [37]
Su *et al.*	**<u>0.134</u>**	**26.98**	**<u>0.125</u>**	**27.77**	**<u>0.130</u>**	**27.16**	CVPR2020 [33]
Vitoria *et al.*	–	25.57	–	–	–	–	WACV2020 [34]
Larsson *et al.*	0.188	24.93	0.183	25.06	0.161	25.72	ECCV2016 [19]
Iizuka *et al.*	0.200	23.63	0.185	23.86	0.146	25.58	ACMToG2016 [15]

Colour Error loss. Our final objective is to generate the (a, b) chrominance channels. To achieve this, the **AB** layer takes $\hat{\mathbf{Z}}$ as input and processes it with a 1×1-`Conv` layer reducing the Q feature maps to 2. These represent the predicted $(\hat{a}, \hat{b}) \in \mathbb{R}^{H \times W \times 2}$ channels obtained by minimising their difference with the real chrominance ones (a, b) as:

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

Combined Loss. We optimise our model for $\mathcal{L}_q + \mathcal{L}_c$. This allows us to generate a plausible colourisation for a same/similar object by exploiting the quantised distribution (learned through $\mathcal{L}_q$) while avoiding the weakness of using $\mathcal{L}_c$ alone, which produces desaturated colours [37].

4. Experiments

To validate our approach, we present extensive experimental results on three benchmark datasets using different evaluation metrics (§ 4.1). Precisely, in § 4.2, we validate and compare our colourisation performance with recent and relevant works [33], also through a large scale user-study. In § 4.3, we evaluate the key components of our approach. In § 4.4, we test colourisation as a method for representation learning. Finally, in § 4.5, we show qualitative examples on legacy black and white images.

4.1. Settings

Datasets. To assess the performance of our approach, we have considered three benchmark datasets coming with different features and colourisation challenges.

***ImageNet*[31]** is widely used as colourisation benchmark. The 1.3M training images (with no labels) were considered for model training for all the following experiments. The ctest10k [19] samples have been used for evaluation.

***COCOStuff*[2]** contains a wide variety of natural scenes with multiple objects present in the 118k images. We used the provided validation split containing 5000 images.

***Places205* [42]** is a scene-centric dataset containing samples 205 different categories. We considered the 20500 validation images.

Note that, COCOStuff and Places205 have been used only for evaluating the colourisation transferability. We do not use the corresponding training sets to learn the model parameters.

Evaluation metrics. To assess the colourisation quality, we followed the experimental protocol proposed in [20] and considered the Peak Signal to Noise Ratio (PSNR) and the the Learned Perceptual Image Patch Similarity (LPIPS) [39] (version 0.1 with VGG backbone).

Implementation Details. We train our network for 20 epochs with a batch size of 32 on the 1.3M ImageNet training samples (with no labels) resized to 224×224. Each image is first projected into the CIELab colourspace, then the resulting L channel is used as the input. The (a, b) channels are considered to compute the $\mathcal{L}_q$ (after quantisation) and $\mathcal{L}_c$ losses. We used the Adam optimiser with a learning rate of 2×10^{-5}.The precise details of the whole UCapsNet architecture are in the supplementary material. Using the PyTorch framework, a single epoch takes about 10 hours on an NVidia Titan RTX.

4.2. State-of-the-art Comparisons

4.2.1 Quantitative Performance

Standard Evaluation Metrics. Table 1 shows the current leaderboard on the three considered datasets. Results show that our approach outperforms all existing solutions in terms of PSNR with a significant margin over methods exploiting labelled data, i.e., we increase the PSNR of [33] by more than 10% on the three datasets on average. LPIPS performance significantly improve with respect to existing methods that do not using semantic labels for training or exploit pre-trained models with supervision (e.g., 0.141 vs 0.202 obtained by [20]). Concerning approaches exploiting labelled data, our method achieves very similar LPIPS per-

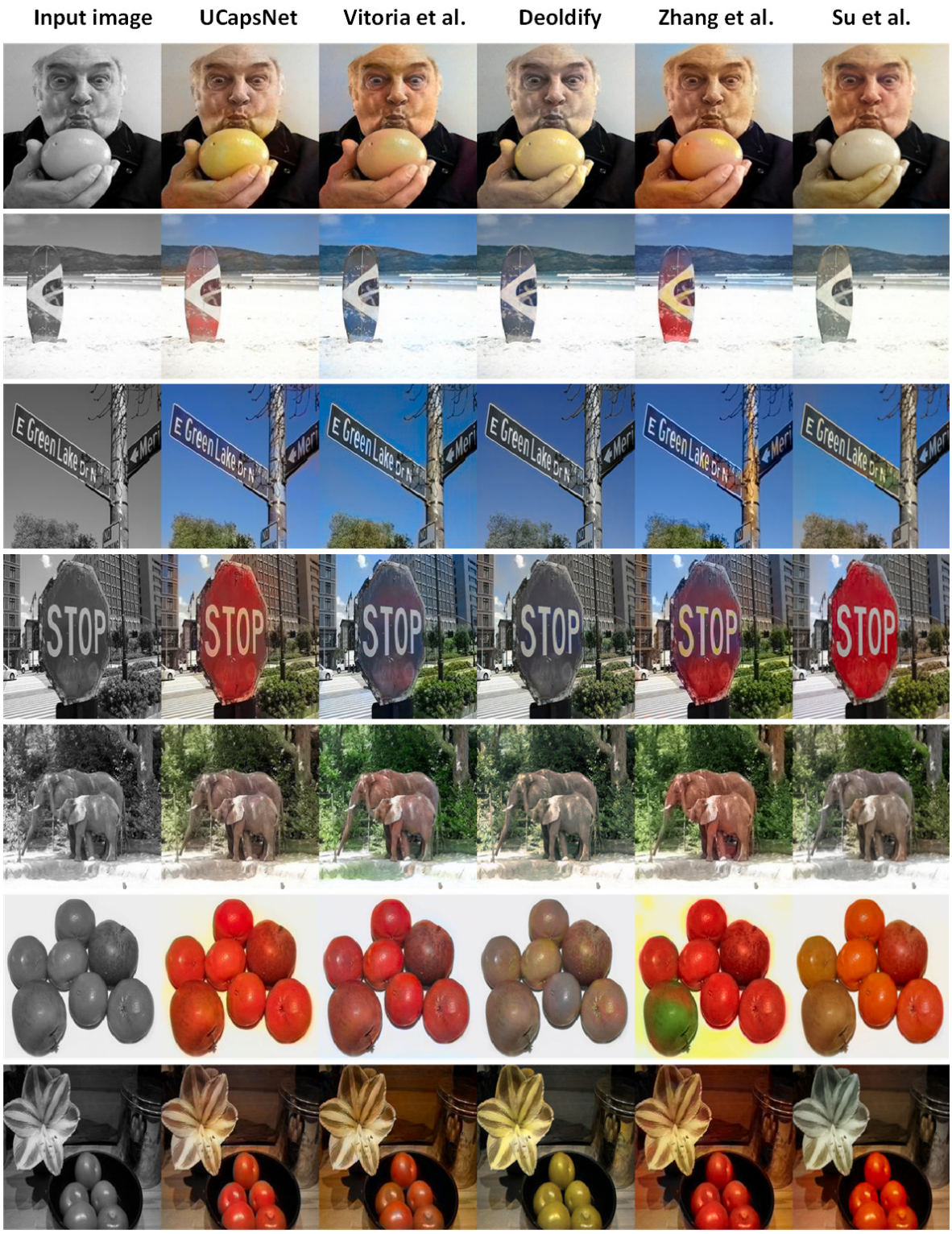

Figure 3: Comparison between state-of-the-art automatic colourisation methods. Our method generates vivid colours that are well defined inside the contours of the entities and contain no splotches. Images are plausible and pleasing for different complex scenes with multiple object instances. Results obtained on COCOStuff dataset.

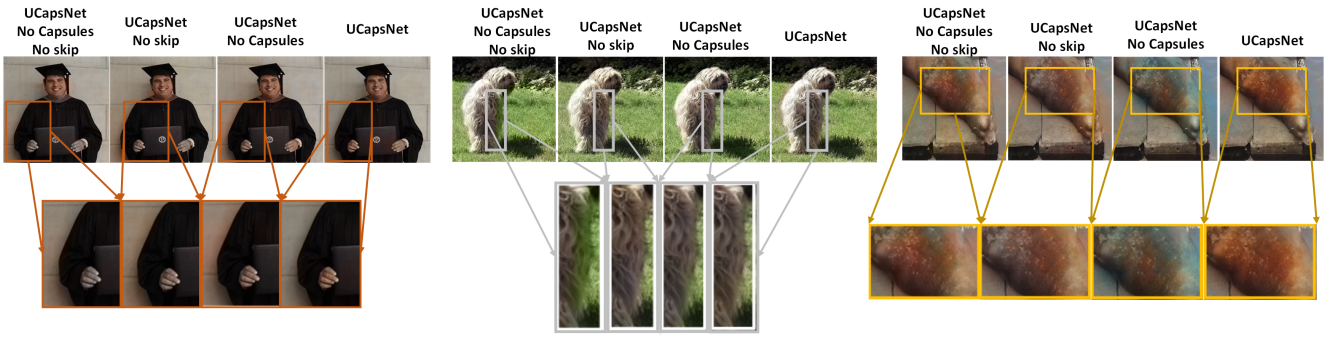

Figure 4: Analysis of the importance of image-level and entity-level features, and their collaboration in UCapsNet on three image samples from COCOStuff. The first row compares the predicted colourisations of a baseline model where capsules and skip connections are not considered (UCapsNet No Capsules No Skip) with three variants were we included (i) capsules (UCapsNet No Skip), (ii) skip connections (UCapsNet No Capsules), and (iii) both together (UCapsNet). In the second row, an expanded view showing detailed differences on the predicted images. These demonstrate the advantage of exploiting both the image and object representations, then enforcing their collaborations to produce a plausible colourisation.

formance to [33] which hinges on a pre-trained model for object detection. Note that for Places205 the performance difference with [33] is only of 0.1%. Such results demonstrate that our approach is able to capture relevant image and entity features to achieve competitive results, without the need of explicit information (e.g., labelled data or dedicated object detection models).

User Study. We conducted a user study to quantify the perceptual realism of the colourisation results obtained with our method in comparison with Vitoria *et al.* [34], DeOldify [1], Zhang *et al.* [37], and Su *et al.* [33]. We randomly selected 200 images from the COCOStuff validation dataset. We show to each participant 20 image pairs composed of our generated image and an image generated by one of the above methods (randomly) and we asked for preference. We collected a total of 3600 votes from 180 participants. Results show that our method is preferred over Vitoria *et al.* [34] (53% vs 47%), DeOldify [1] (64% vs 36%), Zhang *et al.* [37] (54% vs 46%), and Su *et al.* [33] (54% vs 46%). From a qualitative analysis, we noted that images with saturated/vivid colours are preferred even in presence of colour splotches or implausible colours.

4.2.2 Qualitative Performance

Figure 3 compares the colourisation results of our approach and competing solutions on the COCOStuff dataset. In general, we notice that our method provides a consistent object/background separation also reducing the colour blurring on contours thus generating more detailed outputs. Colours look vivid in all the images, and this is more evident when our results are compared with Deoldify [1], Vitoria *et al.* [34] and Su *et al.* [33] (e.g., examples at rows 1, 2, 5, and 7). With respect to Zhang *et al.* [37] (fifth column), our generated colours are better defined inside the contours of the entities and contain no splotches (e.g., examples at rows 3, 4, 6). Finally, it is worth noting UCapsNet provides specific colours entangled with the nature of the entities in the image (e.g., examples at rows 1, 2, 4).

4.3. Ablation Study

We validate the main design choices of our model by analysing the role of the key architecture components, then we evaluate the effects of the considered loss functions.

UCapsNet Variants. To assess the importance of image-level features, entity-level features and their collaboration for the colourisation objective, we start from the model baseline (UCapsNet No Capsules, No Skip) and progressively included our contributions. In Figure 4, we report on the results obtained considering (i) capsules (UCapsNet No Skip), (ii) skip connections (UCapsNet No Capsules) and (iii) both (UCapsNet). The baseline model (UCapsNet No capsules No skip) does not respect object boundaries, e.g., the fur of the dog is partially green as the grass, and generates inconsistent entity colourisations, like the hands of the man which are grey. By adding entity-level features (UCapsNet No Skip), object are recognised and their boundaries respected. No splotches are also present in the results. When skip-connections are included and capsules rejected (UCapsNet No Capsules), the colourisation overflows the object boundaries, thus demonstrating that capsules carry important entity-level features, e.g., the seal image, where the animal shape is not recognised. Such results substantiate the importance of the image-level and entity-level features extraction then promoting their collaboration to generate a well defined colourisation, i.e., UCapsNet results.

Figure 5: Applying UCapsNet to black and white photographs from Henri Cartier Bresson, Ansel Adams, Alinari. Historical archives.

Table 2: Analysis of the importance of the considered losses. Results are computed for UCapsNet optmised by considering the loss functions defined in § 3.4. Best resutls are in bold.

Method	LPIPS↓	PSNR↑
UCapsNet $\mathcal{L}_q$	0.439	28.67
UCapsNet $\mathcal{L}_c$	0.206	30.40
UCapsNet $\mathcal{L}_c + \mathcal{L}_q$	**0.141**	**30.42**

Losses. To evaluate the role of the loss functions composing our optimisation objective, we have trained our model separately considering $\mathcal{L}_q$ and $\mathcal{L}_c$. LPIPS and PSNR performance in Table 2 show that the combination of the two losses improves the results achieved by exploiting either $\mathcal{L}_c$ or $\mathcal{L}_q$. Such a result substantiates the importance of jointly learning a colour distribution as well as a mapping to the chroma channels to generate a plausible colourisation able to handle the multi-modality of appearance of entities.

4.4. Colourisation task as Pretext

In Figure 6, we evaluate the trained model for representation learning. For a fair comparison, we followed the protocol in [37] and evaluated the generalisation capability of the convolutional features learned in the downsampling phase, hence excluded the entity-level representations. At the output of each `DBD`, we stacked a `MaxPool` layer, with equal kernel and stride sizes such that the generated feature dimensionality is below 10k [37]. Then, we added a linear classifier for the ImageNet classes that is trained for 100 epochs. In Figure 6, we compare with methods working on RGB inputs [8, 9, 28, 38], with models trained on grayscale images [37] (like we do) and with models initialized with random/Gaussian weights or using the k-means scheme [18]. Performance obtained with *layer1* output (i.e., $\texttt{DBD}_4$ in our architecture), is in line with the methods based on grayscale input. With increasing depths, i.e., at *layer3* and *layer4* our method reaches competitive results. The classification accuracy obtained with the *layer4* features (i.e., 29%) is in line with existing methods working on an RGB input. This shows that, despite the input handicap, representations obtained as a pretext task carry relevant information to discriminate among different semantic classes.

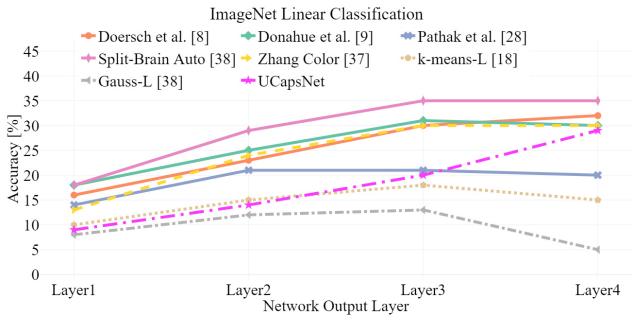

Figure 6: Task Generalisation on ImageNet. We freeze each downsampling layer and evaluate the features extracted at different depths by stacking a classification layer on top of those. Methods working on a greyscale input are shown with dotted lines while approaches considering an RGB input are with solid lines.

4.5. Colourising Legacy Black and White Photos

Our model is trained using generated grayscale images by removing the (a, b) channels from coloured photos. In Figure 5, we show that UCapsNet is able to produce realistic colourisations on real legacy black and white photos, even though the low-level statistics of the such photos are different from the modern-day images used for training.

5. Conclusion

Image colourisation is an instance of a difficult pixel prediction problem in computer vision. We have demonstrated that extracting image-level (through CNN) and entity-level (via capsules layers) features, then enforcing collaboration among them, produces results indistinguishable from real colour photos. Without any prior knowledge, we are able to colourise a greyscale picture respecting the relevant image details and entities differences. Through extensive experiments, we demonstrated that our method outperforms existing colourisation approaches methods that hinge on labelled training data. We have also shown that our model works very well as a pretext task for representation learning, performing strongly compared to other pre-trained colourisation methods.

References

[1] Jason Antic. *A deep learning based project for colorizing ad restoring old images(and videos!)*, 2019.

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

[3] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. In *European conference on computer vision*, pages 126–139. Springer, 2008.

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

[5] Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. Semantic colorization with internet images. *ACM Transactions on Graphics (TOG)*, 30(6):1–8, 2011.

[6] 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.

[7] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 567–575, 2015.

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[9] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

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

[11] Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, and Huang Zhiyong. Image colorization using similar images. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 369–378, 2012.

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

[13] 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.

[14] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 351–354, 2005.

[15] 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.

[16] Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. In *Rendering Techniques*, pages 201–210. Citeseer, 2005.

[17] 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.

[18] Philipp Krähenbühl, Carl Doersch, Jeff Donahue, and Trevor Darrell. Data-dependent initializations of convolutional neural networks. *arXiv preprint arXiv:1511.06856*, 2015.

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

[20] 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.

[21] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694. 2004.

[22] Bo Li, Yu-Kun Lai, and Paul L Rosin. A review of image colourisation. *Handbook Of Pattern Recognition And Computer Vision*, page 139, 2020.

[23] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320, 2007.

[24] 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.

[25] 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.

[26] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

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

[28] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[29] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. *ACM Transactions on Graphics (TOG)*, 25(3):1214–1220, 2006.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmen-

tation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[31] 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.

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

[33] 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.

[34] 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.

[35] Tomihisa Welch, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. *ACM Transactions on Graphics (TOG)*, 21(3):277–280, 2002.

[36] Liron Yatziv and Guillermo Sapiro. Fast image and video colorization using chrominance blending. *IEEE transactions on image processing*, 15(5):1120–1129, 2006.

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

[38] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

[39] 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.

[40] 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.

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

[42] 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.