Learning Diverse Image Colorization

Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh and David Forsyth
Department of Computer Science
University of Illinois at Urbana Champaign
{ardeshp2, jlu23, myeh2, daf} @illinois.edu

Abstract

Colorization is an ambiguous problem, with multiple viable colorizations for a single grey-level image. However, previous methods only produce the single most probable colorization. Our goal is to model the diversity intrinsic to the problem of colorization and produce multiple colorizations that display long-scale spatial co-ordination. We learn a low dimensional embedding of color fields using a variational autoencoder (VAE). We construct loss terms for the VAE decoder that avoid blurry outputs and take into account the uneven distribution of pixel colors. Finally, we develop a conditional model for the multi-modal distribution between grey-level image and the color field embeddings. Samples from this conditional model result in diverse colorization. We demonstrate that our method obtains better diverse colorizations than a standard conditional variational autoencoder model.

1. Introduction

In colorization, we predict the 2-channel color field for an input grey-level image. It is an inherently ill-posed and an ambiguous problem. Multiple different colorizations are possible for a single grey-level image. For example, different shades of blue for sky, different colors for a building, different skin tones for a person and other stark or subtle color changes are all acceptable colorizations (Figure 1). In this paper, our goal is to generate multiple colorizations for a single grey-level image that are diverse and at the same time, each realistic. This is a demanding task, because color fields are not only cued to the local appearance but also have a long-scale spatial structure. Sampling colors independently from per-pixel distributions makes the output spatially incoherent and it does not generate a realistic color field (See Figure 2, output of testing procedure). Therefore, we need a method that generates multiple colorizations while balancing per-pixel color estimates and long-scale spatial co-ordination. This paradigm is common to many ambiguous vision tasks where multiple predictions are desired viz. generating motion-fields from static image [25], synthesizing future frames [27], time-lapse videos [31], interactive segmentation and pose-estimation [1] etc.

A natural approach to solve the problem is to learn a conditional model $P(\mathbf{C}|\mathbf{G})$ for a color field $\mathbf{C}$ conditioned on the input grey-level image $\mathbf{G}$. We can then draw samples from this conditional model $\{\mathbf{C}_k\}_{k=1}^N \sim P(\mathbf{C}|\mathbf{G})$ to obtain diverse colorizations. To build this explicit conditional model is difficult. The difficulty being $\mathbf{C}$ and $\mathbf{G}$ are high-dimensional spaces. The distribution of natural color fields and grey-level features in these high-dimensional spaces is therefore scattered. This does not expose the sharing required to learn a conditional model that encodes diversity. Therefore, we seek feature representations of $\mathbf{C}$ and $\mathbf{G}$ that allow us to build a conditional model.

Our strategy is to represent $\mathbf{C}$ by its low-dimensional latent variable embedding $\mathbf{z}$. This embedding is learned by a generative model such as the Variational Autoencoder (VAE) [14] (See Step 1 of Figure 1). Next, we leverage a Mixture Density Network to learn a multi-modal conditional model $P(\mathbf{z}|\mathbf{G})$ (See Step 2 of Figure 1). Our feature representation for grey-level image $\mathbf{G}$, comprises the features from conv-7 layer of a colorization CNN [30] which encodes spatial structure and per-pixel affinity to colors. Finally, at test time we sample multiple $\{\mathbf{z}_k\}_{k=1}^N \sim P(\mathbf{z}|\mathbf{G})$ and use the VAE decoder to obtain the corresponding colorizations $\mathbf{C}_k$ for each $\mathbf{z}_k$ (See Figure 1). Note that our low-dimensional embedding encodes the spatial structure of color fields and we obtain spatially coherent diverse colorizations by sampling the conditional model.

The contributions of our work are as follows. First, we learn a smooth low-dimensional embedding along with a device to generate corresponding color fields with high fidelity (Section 3, 7.2). Second, we a learn multi-modal conditional model between the grey-level features and the low-dimensional embedding (Section 4) capable of producing diverse colorizations (Section 7.3). Third, we show that our method outperforms the strong baseline of conditional variational autoencoders for obtaining diverse colorizations (Section 7.3, Figure 4).

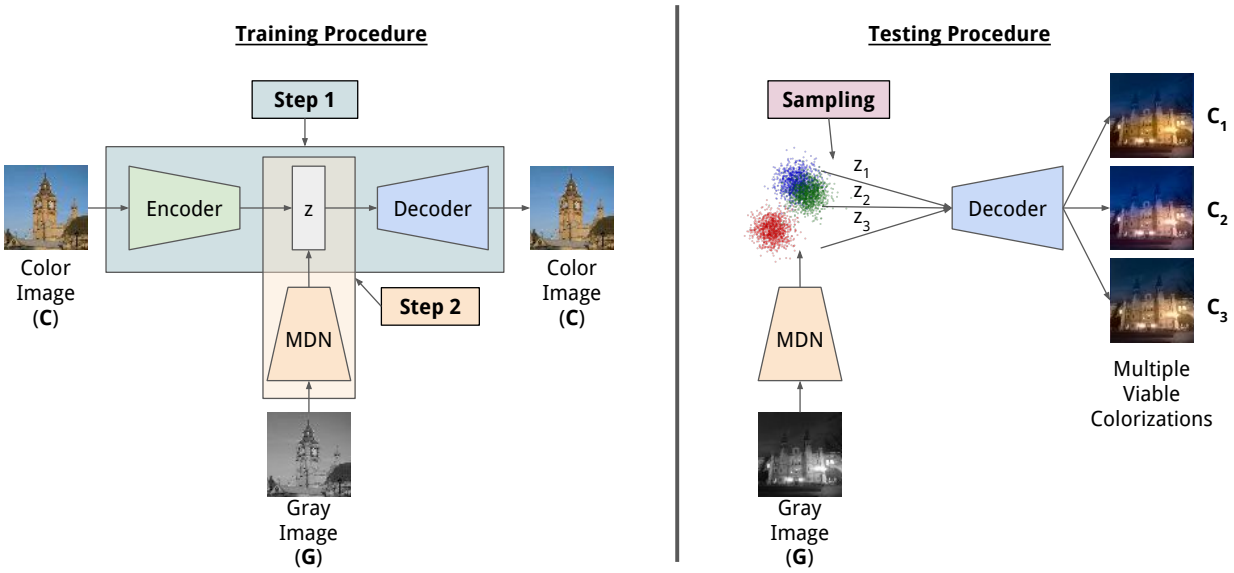
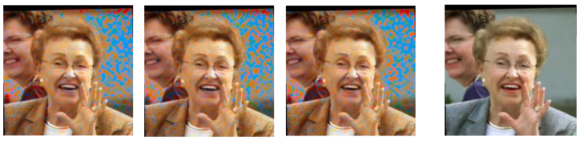


Figure 1: Step 1, we learn a low-dimensional embedding $\mathbf{z}$ for a color field $\mathbf{C}$. Step 2, we train a multi-modal conditional model $P(\mathbf{z}|\mathbf{G})$ that generates the low-dimensional embedding from grey-level feautres $\mathbf{G}$. Finally, we can sample the conditional model $\{\mathbf{z}_k\}_{k=1}^N \sim P(\mathbf{z}|\mathbf{G})$ and use the VAE decoder to generate corresponding diverse color fields $\{\mathbf{C}_k\}_{k=1}^N$.

2. Background and Related Work

Colorization. Early colorization methods were interactive, they used a reference color image [26] or scribble-based color annotations [18]. Subsequently, [4, 3, 5, 20, 11] performed automatic image colorization without any human annotation or interaction. However, these methods were trained on datasets of limited sizes ranging from a few tens to a few thousands of images. Recent CNN-based methods have been able to scale to much larger datasets of a million images [30, 16, 8]. All these methods are aimed at producing only a single color image as output. [3, 30, 16] predict a multi-modal distribution of colors over each pixel. But, [3] performs a graph-cut inference to produce a single color field prediction, [30] take expectation after making the per-pixel distribution peaky and [16] sample the mode or take the expectation at each pixel to generate single colorization. To obtain diverse colorizations from [30, 16], colors have to be sampled independently for each pixel. This leads to speckle noise in the output color fields as shown in Figure 2. Furthermore, one obtains little diversity with this noise. Isola et al. [10] use conditional GANs for the colorization task. Their focus is to generate single colorization for a grey-level input. We produce diverse colorizations for a single input, which are all realistic.

Variational Autoencoder. As discussed in Section 1, we wish to learn a low-dimensional embedding $\mathbf{z}$ of a color field $\mathbf{C}$ alongwith a decoder to generate the color field for any embedding. Kingma and Welling [14] demonstrate that this can be achieved using a variational autoencoder comprising of an encoder network and a decoder network. They derive the following lower bound on log likelihood:



(a) Sampling per-pixel distribution of [30] (b) Ground truth

Figure 2: Zhang et al. [30] predict a per-pixel probability distribution over colors. First three images are diverse colorizations obtained by sampling the per-pixel distributions independently. The last image is the ground-truth color image. These images demonstrate the speckled noise and lack of spatial co-ordination resulting from independent sampling of pixel colors.

$$\mathbb{E}_{z\sim Q}[\log P(\mathbf{C}|\mathbf{z},\theta)] - \mathcal{KL}[Q(\mathbf{z}|\mathbf{C},\theta)\|P(\mathbf{z})] \quad (1)$$

The lower bound is maximized by maximizing Equation 1 with respect to parameters θ. They assume the posterior $P(\mathbf{C}|\mathbf{z},\theta)$ is a Gaussian distribution $\mathcal{N}(\mathbf{C}|f(\mathbf{z},\theta),\sigma^2)$. Therefore, the first term of Equation 1 reduces to a decoder network $f(\mathbf{z},\theta)$ with an L_2 loss $\|\mathbf{C} - f(\mathbf{z},\theta)\|_2$. Further, they assume the distribution $P(\mathbf{z})$ is a zero-mean unit-variance Gaussian distribution. Therefore, the encoder network $Q(\mathbf{z}|\mathbf{C},\theta)$ is trained with a

KL-divergence loss to the distribution $\mathcal{N}(0, I)$. Sampling, $z \sim Q$, is performed with the re-parameterization trick to enable backpropagation and the joint training of encoder and decoder. VAEs have been used to embed and decode Digits [14, 12, 6], Faces [15, 28] and more recently CIFAR images [13, 6]. However, they are known to produce blurry and over-smooth outputs. We carefully devise loss terms that discourage blurry, greyish outputs and incorporate specificity and colorfulness in generated color fields (Section 3).

3. Embedding and Decoding a Color Map

We use a VAE to obtain a low-dimensional embedding for a color field. In addition to this, we also require an efficient decoder that generates a realistic color field from a given embedding. Here, we develop loss terms for VAE decoder that avoid the over-smooth and washed out (or greyish) color fields obtained with the standard L_2 loss.

3.1. Decoder Loss

Specificity. Top-k principal components, $\mathbf{P}_k$, are the directions of projections with maximum variance in the high dimensional space of color fields. Therefore, producing color fields that vary primarily along the top-k principal components provides reduction in L_2 loss at the expense of specificity in generated color fields. To disallow this, we project the generated color field $f(\mathbf{z}, \theta)$ and ground-truth color field $\mathbf{C}$ along top-k principal components. We use $k = 20$ in our implementation. Next, we divide the difference between these projections along each principal component by the corresponding standard deviation σ_k estimated from training set. This encourages changes along all principal components to be on an equal footing in our loss. In addition to this, the residue is divided by standard deviation of the k^{th} component. The standard deviation along the k^{th} component is small and this leads to a high weight on the residue with respect to L_2 loss. Write specificity loss $\mathcal{L}_{mah}$ using the squared sum of these distances and residue.

$$\mathcal{L}_{mah} = \sum_{k=1}^{20} \frac{\|[\mathbf{C} - f(\mathbf{z}, \theta)]^T \mathbf{P}_k\|_2^2}{\sigma_k^2} + \frac{\|\mathbf{C}_{res} - f_{res}(\mathbf{z}, \theta)\|_2^2}{\sigma_{20}^2}$$

$$\mathbf{C}_{res} = \mathbf{C} - \sum_{k=1}^{20} \mathbf{C}^T \mathbf{P}_k \mathbf{P}_k$$

$$\mathbf{f}_{res}(\mathbf{z}, \theta) = \mathbf{f}(\mathbf{z}, \theta) - \sum_{k=1}^{20} \mathbf{f}(\mathbf{z}, \theta)^T \mathbf{P}_k \mathbf{P}_k,$$

The above loss is a combination of Mahalanobis distance [19] between vectors $[\mathbf{C}^T\mathbf{P}_1, \mathbf{C}^T\mathbf{P}_2, \cdots, \mathbf{C}^T\mathbf{P}_{20}]$ and $[f(\mathbf{z}, \theta)^T\mathbf{P}_1, f(\mathbf{z}, \theta)^T\mathbf{P}_2, \cdots, f(\mathbf{z}, \theta)^T\mathbf{P}_{20}]$ with a diagonal covariance matrix $\Sigma = \text{diag}(\sigma_k)_{k=1 \text{ to } 20}$ and an additional residual term.

Colorfulness. The distribution of colors in images is highly imbalanced, with more greyish colors than others. This biases the generative model to produce color fields that are washed out. Zhang et al. [30] address this by performing a re-balancing in the loss that takes into account the different populations of colors in the training data. The goal of re-balancing is to give higher weight to rarer colors with respect to the common colors.

We adopt a similar strategy that operates in the continuous color field space instead of the discrete color field space of Zhang et al. [30]. We independently divide the 'ab' color field (of Lab color space) into a uniform grid with 64 bins each for a and b values. To estimate the distribution of colors in natural images, we compute a histogram over these bins using a representative set for natural images. Specifically, we use images from our train split of Imagenet-val dataset (See Section 7.1) to compute this histogram. The histogram is normalized to sum to 1. For pixel p, we quantize it to obtain its bin and retrieve the inverse of normalized histogram $\frac{1}{H_p}$. $\frac{1}{H_p}$ is used as a weight in the squared difference between predicted color $f_p(\mathbf{z}, \theta)$ and ground-truth $\mathbf{C}_p$ at pixel p.

$$\mathcal{L}_{hist} = \|(H^{-1})^T[\mathbf{C} - f(\mathbf{z}, \theta)]\|_2^2 \tag{2}$$

Gradient. In addition to the above, we also use a first order loss term that encourages generated color fields to have the same gradients as ground truth. Write ∇_h and ∇_v for horizontal and vertical gradient operators, the loss term is as follows:

$$\mathcal{L}_{grad} = \|\nabla_h \mathbf{C} - \nabla_h f(\mathbf{z}, \theta)\|_2^2 + \|\nabla_v \mathbf{C} - \nabla_v f(\mathbf{z}, \theta)\|_2^2 \tag{3}$$

Write overall loss $\mathcal{L}_{dec}$ on the decoder as

$$\mathcal{L}_{dec} = \mathcal{L}_{hist} + \lambda_{mah}\mathcal{L}_{mah} + \lambda_{grad}\mathcal{L}_{grad} \tag{4}$$

We set hyper-parameters $\lambda_{mah} = .1$ and $\lambda_{grad} = 10^{-3}$. The loss on the encoder $\mathcal{L}_{enc}$ (Equation 5) is the same as [14]. We weight this loss by a factor 10^{-4} with respect to the decoder loss. This relaxes the regularization of the low-dimensional embedding, but gives greater importance to the fidelity of color field produced by the decoder. Our relaxed constraint on embedding space does not have adverse effects. Because, our conditional model (Refer Section 4) manages to produce low-dimensional embeddings which decode to natural colorizations (See Figure 4).

$$\mathcal{L}_{enc} = \mathcal{KL}[Q(\mathbf{z}|\mathbf{C})\|\mathcal{N}(0, I)] \tag{5}$$

4. Conditional Model: Grey-level to Embedding

We want to learn a multi-modal (one-to-many) conditional model $P(\mathbf{z}|\mathbf{G})$, between the grey-level image $\mathbf{G}$ and the low dimensional embedding $\mathbf{z}$. Mixture density networks (MDN) model the conditional probability distribution of target vectors, conditioned on the input as a mixture of gaussians [2]. This takes into account the one-to-many mapping and allows target vectors to take multiple values conditioned on the same input vector, providing diversity.

MDN Loss. Here, we formulate the loss function for a MDN that models the conditional distribution $P(\mathbf{z}|\mathbf{G})$. The loss function maximizes the conditional log likelihood $P(\mathbf{z}|\mathbf{G})$. Write $\mathcal{L}_{mdn}$ for the MDN loss, M for the number of components, π_i for the mixture coefficients, μ_i for the means and σ for the fixed spherical co-variance of the GMM. π_i and μ_i are produced by a neural network parameterized by ϕ with input $\mathbf{G}$.

$$\mathcal{L}_{mdn} = -\log P(\mathbf{z}|\mathbf{G}) = -\log \sum_{i=1}^{M} \pi_i(\mathbf{G},\phi)\mathcal{N}(\mathbf{z}|\mu_i(\mathbf{G},\phi),\sigma) \tag{6}$$

It is difficult to optimize Equation 7 since it involves a log of summation over exponents of the form $e^{\frac{-\|\mathbf{z}-\mu_i(\mathbf{G},\phi)\|_2^2}{2\sigma^2}}$. The distance $\|\mathbf{z}-\mu_i(\mathbf{G},\phi)\|_2$ is high when the training commences and it leads to a numerical underflow in the exponent. To avoid this, we pick the gaussian component $m = \arg\min_i \|\mathbf{z}-\mu_i(\mathbf{G},\phi)\|_2$ with predicted mean closest to the ground truth code $\mathbf{z}$ and only optimize that component per training step. This reduces the loss function to

$$\mathcal{L}_{mdn} = -\log \pi_m(\mathbf{G},\phi) + \frac{\|\mathbf{z}-\mu_m(\mathbf{G},\phi)\|_2^2}{2\sigma^2} \tag{7}$$

This min-approximation resolves the identifiability (or symmetry) issue within MDN as we tie a grey-level feature to a component (m^{th} component as above). The other components are free to be optimized by nearby grey-level features. Therefore, clustered grey-level features jointly optimize the entire GMM, resulting in diverse colorizations. In Section 7.3 we show that this MDN-based strategy produces better diverse colorizations than the baseline of CVAE discussed below.

5. Baseline: Conditional Variational Autoencoder

Conditional Variational Autoencoders (or CVAE) condition the generative process of VAE on a specific input. Therefore, sampling from a CVAE produces diverse outputs for a single input. Walker et al. [25] use a vanilla CVAE for diverse motion prediction from a static image. Xue et al. [27] introduce cross-convolutional layers between image and motion encoder in CVAE to obtain diverse future frame synthesis. Zhou and Berg [31] generate diverse timelapse videos by incorporating conditional, two-stack and recurrent architecture modifications to standard generative models.

Recall that, for our problem of image colorization the input to CVAE is the feature representation for grey-level image $\mathbf{G}$ and output is the color field $\mathbf{C}$. Sohn et al. [23] derieve a lower bound on conditional log-likelihood $P(\mathbf{C}|\mathbf{G})$ of CVAE. They show that CVAE consists of training an encoder $Q(\mathbf{z}|\mathbf{C},\mathbf{G},\theta)$ network with KL-divergence loss and a decoder network $f(\mathbf{z},\mathbf{G},\theta)$ with an L_2 loss. The difference with respect to VAE being that the encoder and decoder network both have an additional input $\mathbf{G}$. We compare CVAE to our strategy of using MDN for the problem of diverse colorization (Section 7.3).

6. Architecture and Implementation Details

Notation. Before we begin describing each network architecture, note the following notation. Write $C_a(k,s,n)$ for convolutions with kernel size k, stride s, output channels n and activation a, B for batch normalization, $U(f)$ for bilinear upsampling with scale factor f and $F(n)$ for fully connected layer with output channels n. Note, we perform convolutions with zero-padding and our fully connected layers use dropout regularization [24].

6.1. VAE

Radford et al. propose a DCGAN architecture with generator (or decoder) network that can model complex spatial structure of images [21]. We require our decoder network to model the spatial structure of color fields of similar complexity. Therefore, we model the decoder network of our VAE to be similar to the generator network of Radford et al. [21]. We follow their best practices of using strided convolutions instead of pooling, batch normalization [9], ReLU activations for intermediate layers and tanh for output layer, avoiding fully connected layers except when decorrelation is required to obtain the low-dimensional embedding. The encoder network is roughly the mirror of decoder network, as per the standard practice for autoencoder networks.

Decoder Network. The decoder network accepts a d-dimensional embedding. It performs 5 operations of bilinear upsampling and convolutions to finally output a $64 \times 64 \times 2$ color field (a and b of Lab color space comprise the two output channels). The decoder network can be writ-

ten as, Input: $1 \times 1 \times d \rightarrow U(4) \rightarrow C_{ReLU}(4,1,1024) \rightarrow B \rightarrow U(2) \rightarrow C_{ReLU}(5,1,512) \rightarrow B \rightarrow U(2) \rightarrow C_{ReLU}(5,1,256) \rightarrow B \rightarrow U(2) \rightarrow C_{ReLU}(5,1,128) \rightarrow B \rightarrow U(2) \rightarrow C_{tanh}(5,1,2)$.

Encoder Network. The encoder network accepts a color field of size $64 \times 64 \times 2$ and outputs a $d-$dimensional embedding. Encoder network can be written as, Input: $64 \times 64 \times 2 \rightarrow C_{ReLU}(5,2,128) \rightarrow B \rightarrow C_{ReLU}(5,2,256) \rightarrow B \rightarrow C_{ReLU}(5,2,512) \rightarrow B \rightarrow C_{ReLU}(4,2,1024) \rightarrow B \rightarrow F(d)$.

Note, our input color fields are of resolution 64×64 with two channels. We use $d = 32$ (i.e. 32-dimensional embedding of color fields) for LFW dataset and $d = 64$ for the other two datasets (Section 7.1).

6.2. MDN

The input to MDN are the grey-level features $\mathbf{G}$ from [30] and have dimension $28 \times 28 \times 512$. We use 8 components in the output GMM of MDN. The output layer comprises $8 \times d$ activations for means and 8 softmaxed activations for mixture weights of the 8 components. The MDN network uses 5 convolutional layers followed by two fully connected layers and can be written as, Input: $28 \times 28 \times 512 \rightarrow C_{ReLU}(5,1,384) \rightarrow B \rightarrow C_{ReLU}(5,1,320) \rightarrow B \rightarrow C_{ReLU}(5,1,288) \rightarrow B \rightarrow C_{ReLU}(5,2,256) \rightarrow B \rightarrow C_{ReLU}(5,1,128) \rightarrow B \rightarrow FC(4096) \rightarrow FC(8 \times d + 8)$. Equivalently, the MDN is a network with 12 convolutional and 2 fully connected layers, with the first 7 convolutional layers pre-trained on task of [30] and held fixed.

At test time, we can sample multiple embeddings from MDN and then generate diverse colorizations using VAE decoder. However, to study diverse colorizations in a principled manner we adopt a different procedure. We order the predicted means μ_i in descending order of mixture weights π_i and use these top-k ($k = 5$) means as diverse colorizations shown in Figure 4.

6.3. CVAE

In CVAE, the encoder and decoder both take an additional input $\mathbf{G}$. For fair comparison, we use a network of a similar capacity as MDN to generate a d-dimensional embedding of $\mathbf{G}$ and concatenate it to the encoder and decoder input. This grey-level feature encoder of CVAE can be written as, Input: $28 \times 28 \times 512 \rightarrow C_{ReLU}(5,1,384) \rightarrow B \rightarrow C_{ReLU}(5,1,320) \rightarrow B \rightarrow C_{ReLU}(5,1,288) \rightarrow B \rightarrow C_{ReLU}(5,2,256) \rightarrow B \rightarrow C_{ReLU}(5,1,128) \rightarrow B \rightarrow FC(4096) \rightarrow FC(d)$.

At test time, we feed multiple embeddings to the CVAE decoder along with fixed grey-level input to obtain different colorizations. We feed 256 embeddings and perform a k-means clustering of the predicted color fields into 5 cluster centers. These 5 cluster centers are shown in Figure 4.

7. Results

In Section 7.2, we evaluate the performance improvement by the loss terms we construct for the VAE decoder. Section 7.3 shows the diverse colorizations obtained by our method and we also compare it to the CVAE baseline.

7.1. Datasets

We use three datasets with varying complexity of color fields. First, we use the Labelled Faces in the Wild dataset (LFW) [17] which consists of $13,233$ faces images aligned by deep funneling [7]. Since the face images are aligned, this dataset has some structure to it. Next, we use the LSUN-Church [29] dataset with $126,227$ images. These images are not aligned and lack the structure that was present in the face dataset. They are still images of the same scene category. By that virtue, they are more structured than the images in the wild. Finally, we use the validation set of ILSVRC-2015 [22] with $50,000$ images as our third dataset. These images are the most un-structured of the three datasets. For each dataset, we randomly choose a subset of 1000 images as test set and use the remaining images for training.

Dataset	L2-Loss		Mah-Loss		Mah-Loss + Colorfulness + Gradient	
	All	Grid	All	Grid	All	Grid
LFW	.079	.095	.077	.079	**.067**	**.069**
Church	**.049**	**.049**	.056	.056	.050	.050
Imagenet -Val	**.066**	**.068**	.091	.091	.091	.093

Table 1: For test set, our loss terms show comparable mean absolute error per pixel (wrt ground-truth color field) when compared to the standard L_2 loss on LFW and Church.

Dataset	L2-Loss		Mah-Loss		Mah-Loss + Colorfulness + Gradient	
	All	Grid	All	Grid	All	Grid
LFW	7.20	11.29	6.69	7.33	**2.65**	**2.83**
Church	4.9	4.68	6.54	6.42	**1.74**	**1.71**
Imagenet -Val	10.02	9.21	12.99	12.19	**4.82**	**4.66**

Table 2: For test set, our loss terms show better weighted absolute error per pixel (wrt ground-truth color fields) when compared to L_2 loss on all the datasets.

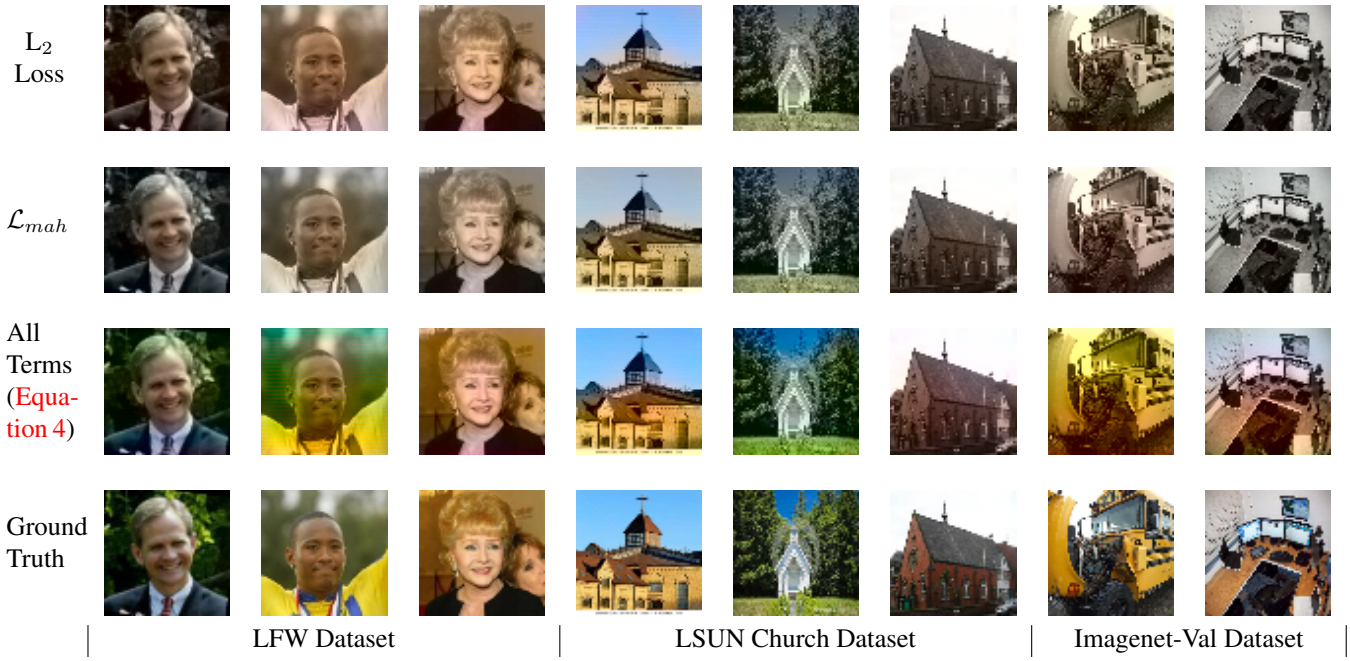


Figure 3: Qualitative results with different loss terms for the VAE decoder network. Top or 1^{st} Row uses only the L_2 loss, 2^{nd} Row uses $\mathcal{L}_{mah}$, 3^{rd} uses all the loss terms: mahalanobis, colorfulness and gradient (Refer $\mathcal{L}_{dec}$ of Equation 4) and last row is the ground-truth color field. These qualitative results show that our loss terms generate better quality color fields as compared to standard the L_2 error for VAE decoders.

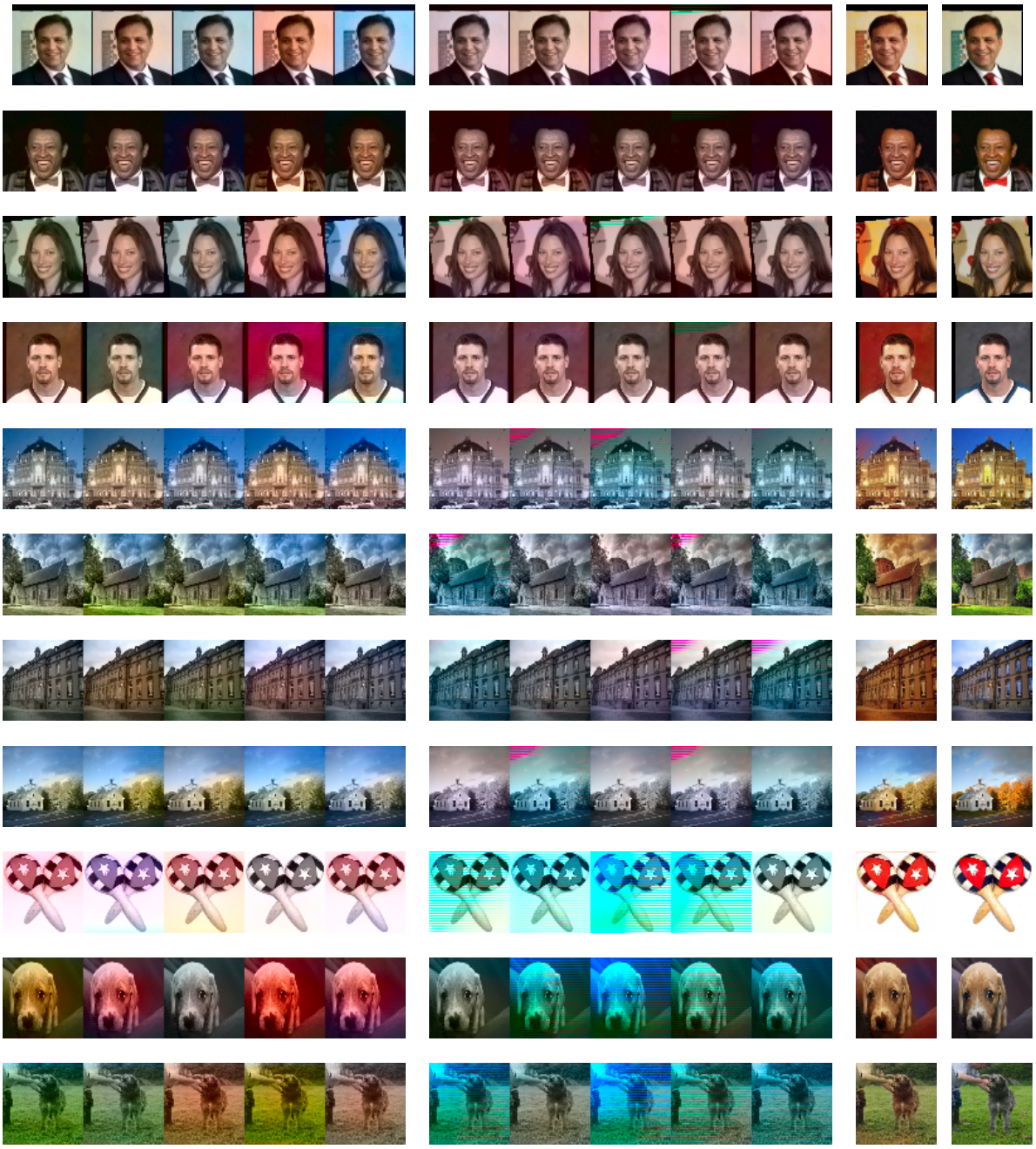
7.2. Effect of Loss terms on VAE Decoder

We train VAE decoders with: (i) the standard L_2 loss, (ii) the specificity loss $\mathcal{L}_{mah}$ of Section 3.1, and (iii) all our loss terms of Equation 4. Figure 3 shows the colorizations obtained for the test set with these different losses. To achieve this colorization we sample the embedding from the encoder network. Therefore, this does not comprise a true colorization task. However, it allows us to evaluate the performance of the decoder network when the best possible embedding is available. Figure 3 clearly demonstrates that for all datasets using all our loss terms provides better colorizations compared to standard L_2 loss. Note, with $\mathcal{L}_{mah}$ the face images in the second row have more contained skin colors as compared to the first row. This shows subtle the benefits obtained from the specificity loss.

In Tables 1, 2 we compare the mean absolute error and mean weighted absolute error per-pixel with respect to the ground-truth for different loss terms. The weighted error uses the same weights as colorfulness loss (Section 3.1). We compute the error over: 1) all pixels (All) and 2) over a 8×8 uniformly spaced grid in the center of image. We compute error on a grid to mitigate the averaging effect on errors over entire image. On the absolute error metric of Table 1, we outperform standard L_2 error metric on LFW and show comparable performance on Church with all our loss terms. Note unlike L_2 loss, we do not specifically train for this absolute error metric and still achieve reasonable performance with our loss terms. On the weighted error metric of Table 2, our loss terms outperform the standard L_2 error on all datasets.

7.3. Comparison of diverse colorizations of MDN and CVAE

In Figure 4, we compare the diverse colorizations generated by our strategy using MDN (Sections 3, 4) and the baseline method using CVAE (Section 5). Qualitatively, we observe that our strategy generates better quality diverse colorizations which are each, realistic. In Figure 5, we plot the error-of-best (i.e. pick the diverse colorization with minimum error to ground-truth colorization) vs. the variance of diverse colorizations for different datasets. Note that, MDN reliably produces lower error-of-best with comparable variance compared to a CVAE. We believe this is the result of our $\min$-approximation in the MDN loss (Section 4). Since, the MDN model cannot improve conditional likelihood by placing means on top of one another. Therefore, it preserves diversity. In contrast, a CVAE can radically distort $\mathbf{G}$ to get tight low diversity clusters of $\mathbf{C}$ which also give high conditional likelihood.



(a) Ours (b) CVAE (c) Zhang et al.[30] (d) GT

Figure 4: Diverse colorizations from our MDN-based method are compared to the CVAE baseline, Zhang et al. [30] and ground-truth. We obtain better diverse colorizations than the CVAE baseline. Our colorization may miss the finer details of [30] but we provide diverse colorizations which they do not. Note, our performance is satisfactory on LFW and Church dataset (Row 1-8). Our diverse colorizations have different background, skin-tone, grass, sky, building color etc. However, Imagenet-Val dataset is un-structured and VAEs cannot generate the color fields accurately (Row 9-11). Therefore, we see degraded performance on Imagenet-Val. CVAE colorizations have artifacts since the low dimensional embedding is sampled randomly, which our MDN-based method avoids. Additional results in Figures 6 (LFW), 7 (Church) and 8 (Imagenet-val).

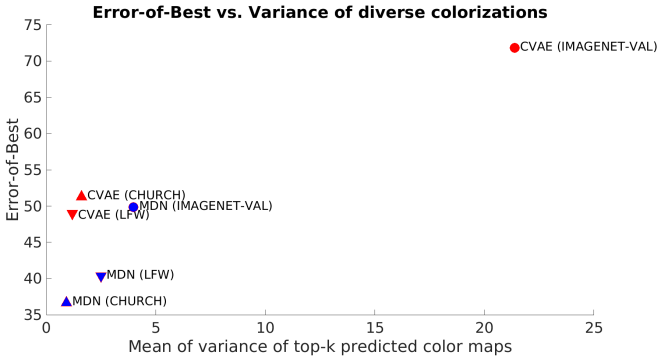


Figure 5: For every dataset, we obtain lower error-of-best to ground-truth using MDN. This shows our MDN-based method generates color fields closer to ground-truth with variance comparable to the CVAE.

Acknowledgements. We thank Arun Mallya and Jason Rock for useful discussions and suggestions.

References

[1] D. Batra, P. Yadollahpour, A. Guzmn-Rivera, and G. Shakhnarovich. Diverse m-best solutions in markov random fields. In *ECCV (5)*, volume 7576 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2012. 1

[2] C. M. Bishop. Mixture density networks, 1994. 4

[3] G. Charpiat, M. Hofmann, and B. Schölkopf. Automatic image colorization via multimodal predictions. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pages 126–139, Berlin, Heidelberg, 2008. Springer-Verlag. 2

[4] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. *ICCV*, abs/1605.00075, 2015. 2

[5] A. Deshpande, J. Rock, and D. A. Forsyth. Learning large-scale automatic image colorization. In *ICCV*, pages 567–575. IEEE Computer Society, 2015. 2

[6] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In D. Blei and F. Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1462–1471. JMLR Workshop and Conference Proceedings, 2015. 3

[7] G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller. Learning to align from scratch. In *NIPS*, 2012. 5

[8] S. Iizuka, E. Simo-Serra, and H. 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 (Proc. of SIGGRAPH 2016)*, 35(4), 2016. 2

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 4

[10] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 2

[11] J. Jancsary, S. Nowozin, and C. Rother. Loss-specific training of non-parametric image restoration models: A new state of the art. *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, pages 112–125, 2012. 2

[12] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc., 2014. 3

[13] D. P. Kingma, T. Salimans, and M. Welling. Improving variational inference with inverse autoregressive flow. *CoRR*, abs/1606.04934, 2016. 3

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 1, 2, 3

[15] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems 28*, pages 2539–2547. Curran Associates, Inc., 2015. 3

[16] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[17] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. *Labeled Faces in the Wild: A Survey*, pages 189–248. Springer International Publishing, Cham, 2016. 5, 10

[18] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, Aug. 2004. 2

[19] P. C. Mahalanobis. On Tests and Measures of Groups Divergence. *International Journal of the Asiatic Society of Benagal*, 26, 1930. 3

[20] Y. Morimoto, Y. Taguchi, and T. Naemura. Automatic colorization of grayscale images using multiple images on the web. In *SIGGRAPH 2009: Talks*, SIGGRAPH '09, New York, NY, USA, 2009. ACM. 2

[21] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 4

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5, 12

[23] K. Sohn, X. Yan, and H. Lee. Learning structured output representation using deep conditional generative models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3483–3491, Cambridge, MA, USA, 2015. MIT Press. 4

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014. 4

[25] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, 2016. 1, 4

[26] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *SIGGRAPH*, 2002. 2

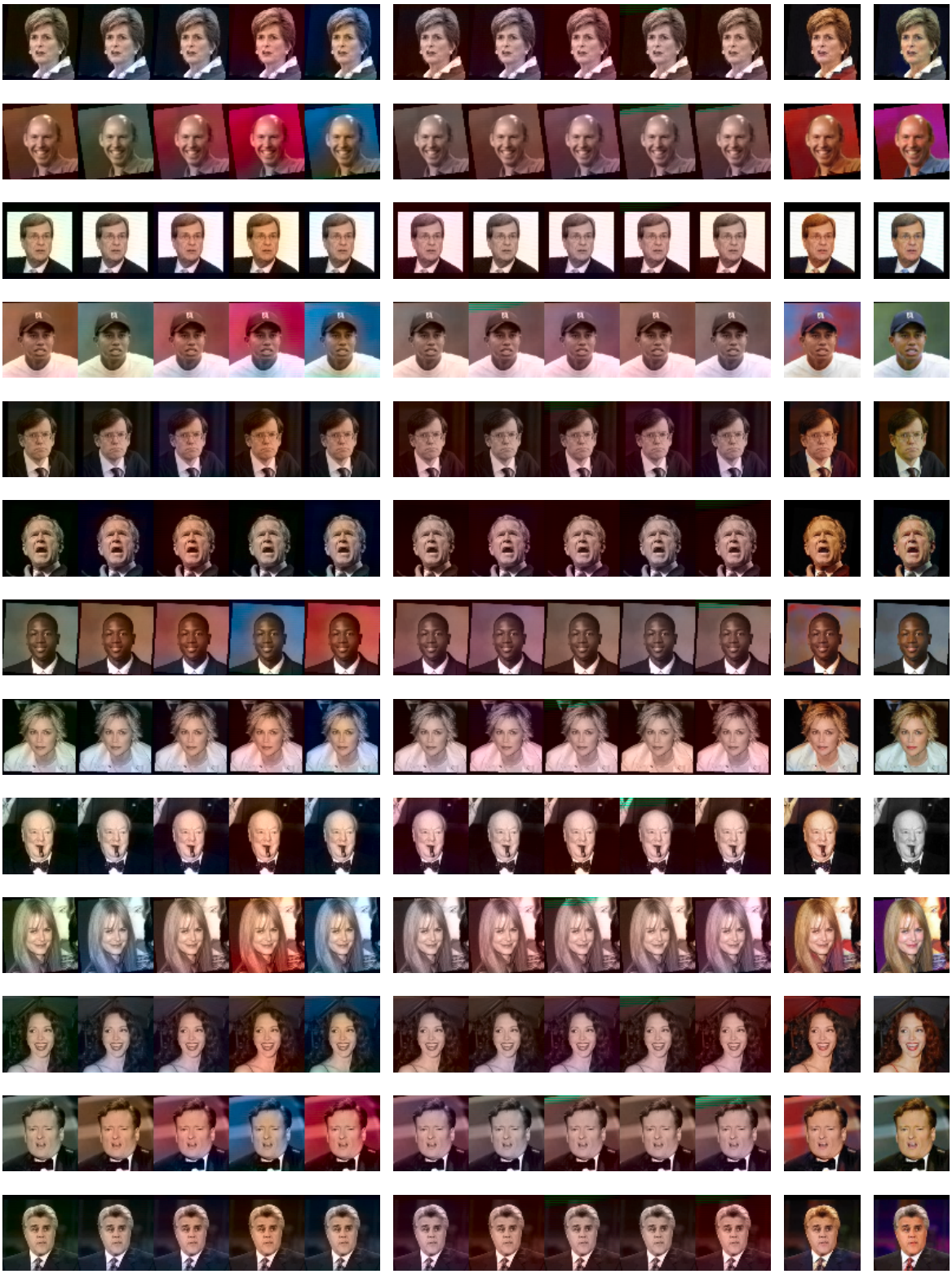
[27] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *CoRR*, abs/1607.02586, 2016. 1, 4

[28] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 776–791, 2016. 3

[29] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. 5, 11

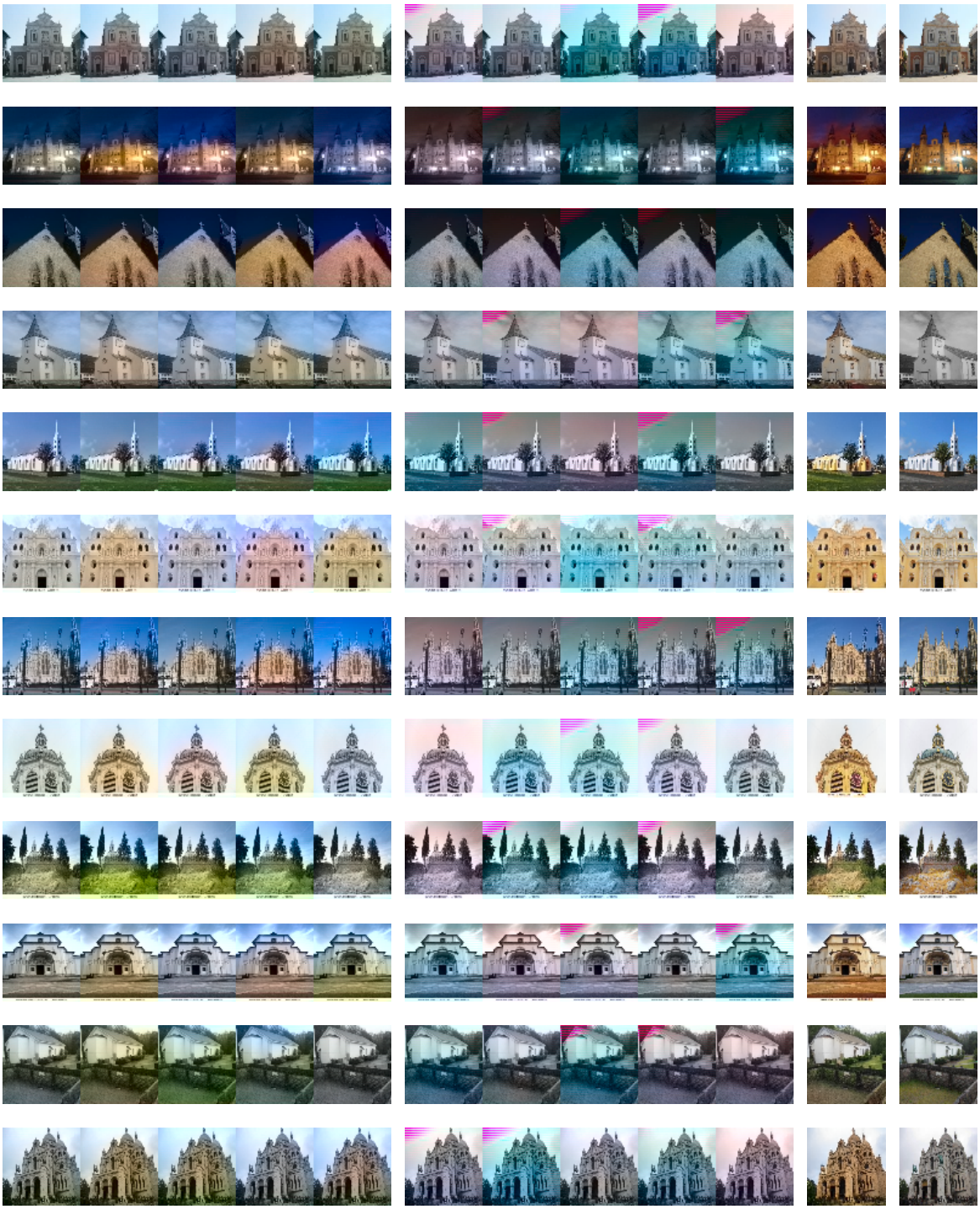
[30] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. *ECCV*, 2016. 1, 2, 3, 5, 7, 10, 11, 12

[31] Y. Zhou and T. L. Berg. *Learning Temporal Transformations from Time-Lapse Videos*, pages 262–277. Springer International Publishing, Cham, 2016. 1, 4



(a) Ours (b) CVAE (c) Zhang et al. [30] (d) GT

Figure 6: Additional results for diverse colorizations from our MDN-based method vs. the CVAE baseline, Zhang et al. [30] and ground-truth on LFW dataset [17].



(a) Ours (b) CVAE (c) Zhang et al. [30] (d) GT

Figure 7: Additional results for diverse colorizations from our MDN-based method vs. the CVAE baseline, Zhang et al. [30] and ground-truth on LSUN Church dataset [29].

(a) Ours (b) CVAE (c) Zhang et al. [30] (d) GT

Figure 8: Additional results for diverse colorizations from our MDN-based method vs. the CVAE baseline, Zhang et al. [30] and ground-truth on Imagenet-val dataset [22].