Video Colourization using Generative Adversarial Networks and Deep Colour Propagation: Final Report

G110 (s1875880, s1879262, s1858332)

Abstract

The purpose of our project is to explore the application of black and white image colourization on videos using Conditional Generative Adversarial Networks (cGANs) and Video Colour Propagation Networks. Initially, we will implement a GAN that transforms a given greyscale image to a colour one. This GAN will be trained using colour images that are first transformed to greyscale. Regarding videos, each of their frames will be colourized using this methodology. Then we will apply a framework for colour propagation using Bilateral and Spatial Networks between the frames of the video to allow for the colourization of consecutive frames in a video. Finally, we will combine the two techniques for a complete video colourization framework.

1. Introduction

In the paper by Isola *et al.* (2017) there are several examples of Image-to-Image translation, including labels to images, edges to images, aerial views to maps and black and white to colour. All these problems are underlined by the same setting which is predicting pixels from other pixels. Nazeri *et al.* (2018) have an implementation of the colourization algorithm by Isola *et al.* and we will be using that implementation for our project.

The popular choice for image prediction problems are Convolutional Neural Networks (CNNs) which learn to minimize a loss function. CNNs require considerable manual effort in determining loss functions (Isola et al., 2017). On the other hand Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) automatically learn a loss function based on a given high-level goal. This process happens by pitting a generative model against a discriminative model where the generative model tries to learn how to generate output which the discriminative can't distinguish from the original. That's why our focus will shift on GANs.

Colourizing independent images is the first part of our project. The second part is finding a way to propagate the colouring information along consecutive frames in order to colour whole frame sequences effectively. While we have stumbled upon several examples of such networks (Meyer et al., 2018), (Xia et al., 2016), (Jampani et al., 2016) we have decided to focus on the latter due to immediate availability of their code. They implemented what they called Video Propagation Networks (VPN) which combine temporal bilateral networks for video adaptive filtering and spatial networks for feature refinement.

Our goal is to combine the Pix2Pix software implemented by Nazeri *et al.* (2018) to convert a set of greyscale frames into RGB and use those as references in a VPN (Jampani et al., 2016) in order to have a complete system that colourizes a fully greyscale video.

In Section 2 we describe the dataset and task that we explored, as well as the preprocessing we applied on the dataset and the evaluation methodologies we used. In Section 3 the two parts of the Pix2Pix software and VPN are described along with the way we implemented their core algorithms to create the whole system. Section 4 describes and displays the experiments we conducted using our complete model. The results of these experiments and comparisons between them are presented in section 5. Section 6 summarizes similar published work in video colour propagation which helps with the understanding of our own task. The final Section 7 contains a summary of what we have learned from the experiments and analyses the outcome of our task in relation to the overall research questions and objectives.

2. Data set and task

For our project we used the DAVIS (Densely Annotated VIdeo Segmentation) dataset (Pont-Tuset et al., 2017) (Perazzi et al., 2016). This dataset consists of high quality, full high definition videos of specific scenes with moving objects in them. Such videos include a walking bear, a moving bus, people crossing the street etc. The videos are split into frames and the average frames per video are 70. In the website of the dataset there is a lower resolution available for download at 854 by 480 pixels. We decided to use the lower resolution dataset due to time and computational restrictions. The dataset was also split into two portions, one for training and validation and one for testing.

While the original purpose of this dataset was to be used in video frame annotation challenges, we thought that the videos contain the necessary amount of movement and complexity to allow us to produce sufficient results from experimenting on them in regards to colourization and colour propagation.

The preprocessing we applied on the dataset was to reduce the resolution quality from 854 by 480 pixels to 256 by 256.

This was done in order to allow the Pix2Pix software associated with the paper by Isola *et al.* (2017) to receive them as input. We didn't need to convert the frames to greyscale because the software was performing the transformation by itself in the training.

The same preprocessing was applied on the test portion of the dataset and also the RGB to greyscale transformation because this is not arbitrarily done by the software during testing.

We defined our task as an implementation and combination of the work of Isola *et al.* as implemented by Nazeri *et al.* (2018) and the work of Jampani *et al.* (2016). Initially we wanted to set up the code of both parts to work for our chosen dataset, the DAVIS video dataset. We decided to conduct three experiments, one on each part and one on their combination. The first two experiments would be a colourization procedure of the DAVIS dataset using each part individually. The Pix2Pix software would colourize all the individual video frames of the DAVIS dataset while the Video Propagation Network (VPN) would colourize the videos based on the first original colour video frame. The final experiment would first use the Pix2Pix software to colourize the initial frame of the video which then would be used as an input to the VPN along with the rest frames in greyscale.

Our basis of evaluation and comparison are the Root Mean Square Error (RMSE) (Barnston, 1992) and Peak Signal-to-Noise Ratio (PSNR) measures. RMSE is the standard deviation of the prediction errors or residuals. The lower the value of RMSE, the better. PSNR computes the ratio between the maximum power of a signal and the power of noise between two images. It is measured in decibels and it is used as a quality measurement for compression of images. For the images we used, PSNR is reported against each channel of the YCbCr colour space. The higher the PSNR value, the better. These two error metrics are generally used for image comparison and both of them are defined via the Mean Squared Error.

3. Methodology

Our project consists of two applications of recent papers in image-to-image translation (Isola et al., 2017) and video colour propagation (Jampani et al., 2016). Each one completes one part of our proposed video colourization framework.

3.1. Pix2Pix

The Pix2Pix software by Nazeri *et al.* (2018) generalized the colourization procedure using a conditional Deep Convolutional Adversarial Network (DCGAN). The network architecture is based on U-Net (Ronneberger et al., 2015), a network and training strategy that relies on the use of data augmentation that allows for more efficient use of the annotated images. The model is symmetric with equal number of encoding and decoding units depending on the image size. In our case, for 256 by 256 pixels, 8 encoding and decoding units are used. There are two paths in the network, a contracting path for downsampling and an expansive path for upsampling. Convolutional layers of size 4 by 4 are used, with stride 2 for downsampling and followed by batch normalization and Leaky-ReLU activation function. The number of channels is doubled after each step. For upsampling, same sized transposed convolutional layers are used with stride 2 which are concatenated with the mirrored activation maps from the contracting path and then followed by batch normalization and ReLU activation function. The final layer is a 1x1 convolution equivalent to a cross-channel parametric pooling layer and the tanh function is used for it. The network architecture is presented in Figure 1, copied from Nazeri *et al.* (2018).

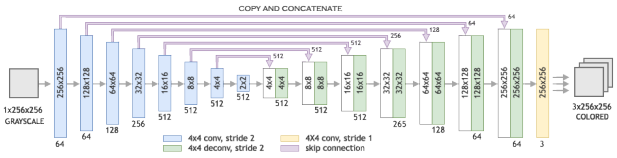


Figure 1. Generator architecture of the image colourization model, copied from Nazeri *et al.* (2018).

The discriminator is a GAN architecture with a similar contracting path as the above architecture where all layers are followed by batch normalization and Leaky-ReLU activation. A sigmoid function is applied after the last layer to return probability values of 70 by 70 patches of the decision of the input being real or fake. The network output is the average of the probabilities.

We followed their GitHub instructions to install the Pix2Pix software. The datasets used by Nazeri *el al.* are different from the one we used. Each dataset has its own class in the code. We created a new class based on the originals to accommodate for the dataset we used and adjusted it to be able to also receive smaller input images of size 32 by 32 pixels, apart from the standard 256 by 256, for the initial training phase. The classes contain three methods, one for creating the generator, one for creating the discriminator and one for creating the dataset.

3.2. Video Propagation Networks

The Video Propagation Network (VPN) (Jampani et al., 2016) is composed of two components. First, a temporal Bilateral Network (BNN) performs image-adaptive spatio-temporal dense filtering to allow dense connection of all pixels from current and previous frames in a video and to propagate associated pixel information to the current frame. In this network, two Bilateral Convolution Layers (BCL) with 32 filters each, filter the input video sequence and its corresponding predictions. Their 32 dimensional outputs are concatenated and passed through a ReLU before being filtered again by the same two BCL layers. The second output is reduced to a 1 by 1 spatial filter. Then, a spatial Convolutional Neural Network (CNN) is applied on the

output of the BNN to refine and predict the present video frame. According to the authors, VPNs can be used to propagate colour which is a continuous information content across video frames, they don't need future frames and can be used for online video analysis, they can efficiently handle large number of input frames and they can be trained end-to-end with good runtimes. The VPN architecture is presented in Figure 2, copied from Jampani *et al.* (2016).

We used the code accompanying the VPN publication from the author's GitHub. Initially we extracted features for each video frame from the videos in the DAVIS dataset using `prepare_feature_data.py`. These features are in the YCbCr colour space. The authors provide a pre-trained colour propagation model which we get by using `get_color_models.sh`. To perform colour propagation using the model and the features we extracted we use `do_color_propagation.py`. This is performed in two stages. In the first we obtain the initial colour propagated videos using BNN-Identity and in the second we perform VPN colour propagation using the colour results of the first model. Each time

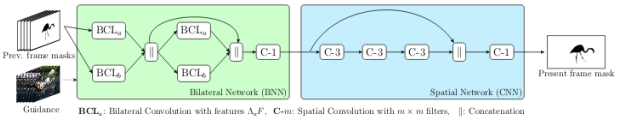
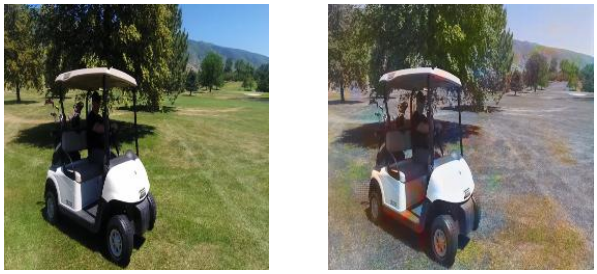


Figure 2. Computation Flow of Video Propagation Network, copied from Jampani *et al.* (Jampani et al., 2016).

4. Experiments

Our experimentation procedure was divided among the two parts of the video colourization system we implemented. Initially we trained the Pix2Pix software with the DAVIS dataset scaled down to 256 by 256 pixels resolution, which was the dataset we decided to use in our project. Due to issues in this initial experiment we decided to use a pretrained version of the Pix2Pix software which we would then use to test the colourization of individual frames from videos in the DAVIS dataset. For the next experiment we moved to the Video Propagation Network (VPN). We implemented the VPN as provided by the authors of the paper proposing it (Jampani et al., 2016) and used the DAVIS dataset as input to get the initial results. We used both the 854 by 480 original pixel resolution and our resized version of 256 by 256 that matched the original dataset used in Pix2Pix. After successfully implementing and testing both parts of the system we proceeded to our final experiment which was to colourize the first frames of the DAVIS videos using Pix2Pix and then passing them as inputs in the VPN along with the rest frames of the videos converted to greyscale in order for it to colourize the whole sequence of them.



(a) Original frame (b) Colourized frame

Figure 3. Pix2Pix trained on DAVIS test output. The original image on the left and the produced one on the right.

4.1. DAVIS trained Pix2Pix

As an initial experiment, we trained a model using the Pix2Pix software with the DAVIS dataset. The experiment run for 55 epochs on the MLP GPU cluster provided, which completed its cycle in 18 hours. The results of the experiment were examined in both quantitative and qualitative manner and we decided that the model's performance was not sufficient enough for further usage. It colourized very small portions of the test frames which were insufficient for this project as it can be seen in Figure 3. The model correctly colourizes the tree behind the kart while ignoring the rest of the scene. A small portion of the grass on the bottom right corner is also colourized. The kart is black and white therefore no effort was required to colourize it.

The reason why the model produced bad results is the training data as well as training time. The pretrained model was trained on a different dataset (Places365 (Zhou et al., 2018)) which had around ten million different images which gave much more variety of "knowledge" to the model. On the other hand, the DAVIS dataset has just around 3470 and 50 categories. Each category has frames that are related, hence less variety than the Places365 dataset. Moreover, we decided that our experiments regarding Pix2Pix would consider videos as individual frames, so a picture colourization would be a better approach rather than videos as explained in the next experiment.

4.2. Pretrained Pix2Pix

As a first experiment we decided to use the pretrained Pix2Pix software to colourize each frame of a video individually. The purpose of this experiment was to provide frames for comparison with the colourized frames produced from the Video Propagation Network in order to show the improvement of colour propagation over individual colourization.

The software was trained on the Places365 dataset (Zhou et al., 2018) which consists of ten million scene photographs.

In figure 4 we present the original first frame of the elephant video as well as the colourized frame from the Pix2Pix software. The first frame of the video is very well colourized

and the result of the software is sufficiently similar to the original frame.

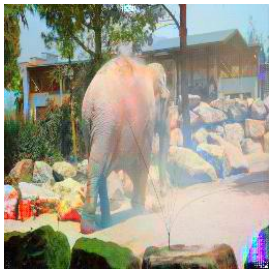
For subsequent frames there are inconsistencies and lack of coherence in the colourization. Notably, there are inconsistencies in the colours on the body of the elephant, which could be caused due to the fact that it is the moving subject of the video. Specifically there are brown patches and faded colours as presented in figure 5.

(a) Original frame 0

(b) Colourized frame 0

Figure 4. Comparison of the first frame of the video in the original and the Pix2Pix output.

(a) Colourized frame 6



(b) Colourized frame 8

Figure 5. Comparison of the 6th and 8th frame of the video as produced by Pix2Pix. The colour of the elephant is not consistent.

4.3. Video Propagation Networks

Video Propagation Network (VPN) (Jampani et al., 2016) is a technique that propagates structured information of a frame to the next. VPN can be used in a variety of video tasks that require propagation of any type of information content between frames, such as video object segmentation and semantic video segmentation. In our project we used VPN to propagate colour in greyscale videos frames.

The authors of VPN (Jampani et al., 2016) provided a pretrained model on the DAVIS dataset (Pont-Tuset et al., 2017). The frames of the dataset that were used for training were 854 by 480 pixels. In our project, the images used as initial frames of the greyscale videos are 256 by 256.

Initially, we tested if there is any loss on the quality of the colour propagation when using the model trained on 854 by 480 to colourize 256 by 256 videos. We colourized the same video twice - with native 854 by 480 resolution and with resized 256 by 256 resolution. The resize was applied to each frame using the resize method of the *Image* module from the *Pillow* Python library using a high-quality downsampling filter, called *ANTIALIAS*.

Table 1 shows the PSNR and RMSE of the native, 854 by 480 video and the resized, 256 by 256 video. There is an insignificant difference of 1.2371 PSNR and 1.3001 RMSE between the results of the two videos. Figure 6 shows the colourization of the breakdancer video with the original resolution and the resized.

	854x480	256x256
PSNR	27.9994	26.7621
RMSE	13.2921	14.5922

Table 1. Comparing the results of the colour propagation of the native video resolution, 854 by 480, and the resized video, 256 by 256.

(a) Original 854 by 480 frame 0

(b) Colourized 854 by 480 frame 6

(c) Colourized 256 by 256 frame 0

(d) Colourized 256 by 256 frame 6

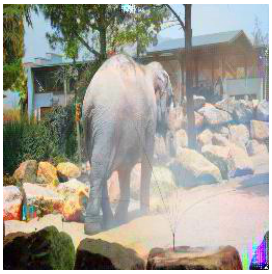
Figure 6. Comparison of the color propagation on 854 by 480 video vs 256 by 256 video.

We concluded that the resize operation does not affect the quality of the colour propagation. The model is able to propagate information on videos that have different resolution than the one used for training.

4.4. Our system (VPN with Pix2Pix)

Our approach focuses on the combination of the two described methods, to achieve a fully automated video colourization system. VPN requires the first frame of the sequence to be the original, coloured one. In our approach, the initial frame is colourized by Pix2Pix instead of providing the original first frame of the video.

The results of our approach are shown in Figure 7, and are better than the colourized frames of Pix2Pix shown in Figures 5. Frame 8 as colourized by Pix2Pix has some unwanted artifacts on the rocks, while in the same frame as colourized by our approach, the result is more natural.



(a) Initial frame colourized by Pix2Pix

(b) Colourized frame 6

(c) Colourized frame 8

(d) Colourized frame 10

Figure 7. Results of colour propagation with VPN and the initial frame colourized by Pix2Pix.

(a) Initial frame colourized by Pix2Pix

(b) Colourized frame 6 with initial Pix2Pix frame

(c) Original initial frame

(d) Colourized frame 6 with original initial frame

Figure 8. Results of our system (a,b) when the initial frame is badly colourized compared to the original VPN (c,d).

The disadvantage of our approach is that when parts of the initial frame are not colourized correctly, these will be propagated to the rest of the video. For example, in Figure 7, the shadow on the bottom right corner was colourized purple by Pix2Pix and this was propagated to the rest frames.

Figure 8 shows the results of our approach when the initial frame is colourized imperfectly by Pix2Pix.

5. Results

	Pix2Pix	VPN	Our system
PSNR	18.6936	26.7621	21.9461
RMSE	31.6966	14.5922	24.6881

Table 2. Performance of the three models on the data-set with the average Peak Signal to Noise Ratio (PSNR) and Root Mean Square Error (RMSE). RMSE lower the better. PSNR higher the better.

Table 2 shows the performance of the three models on our test set. As expected, the original Video Propagation Network (VPN) performed better than the other two since it receives the original coloured frame as the first input while keeping the colours consistent. Pretrained Pix2Pix produces the worst result on the test set. This is because all frames are colourized individually, some frames produce worst result than others, hence influencing the average performance of the model.

The results produced by our system are much better than the individually processed frames by Pix2Pix with a positive 3.25 PSNR and a negative 7 RMSE difference. Moreover, the system is greatly affected by the result of the Pix2Pix software since it takes its output as the first coloured frame input. The overall quality of the video produced is much better than Pix2Pix since it keeps the colours consistent throughout the video frames. For example, in figure 7 the elephant is colourized grey instead of brown but the same colour is kept for the rest of the frames which makes the viewing experience more pleasant.

		VPN	Our system
Elephant	PSNR	30.0427	25.7217
	RMSE	8.0887	13.2605
Swan	PSNR	29.2156	17.2720
	RMSE	9.3316	35.1847

Table 3. Results of VPN and our system on colourization of the elephant and the black swan videos.

As a final quantitative evaluation we tested the VPN and our system with just one video each time. For the first test we used the elephant video which was colourized very well by Pix2Pix and for the second we used the swan video whose colourization was imperfect. The VPN results show the metrics when the original video is given as input while our system shows the metrics when the video given as input is colourized by Pix2Pix. The purpose of this comparison is to show that if the results of the Pix2Pix software are good then the subsequent VPN will also produce good result based on that Pix2Pix output. It is indicative from the results that the Pix2Pix performance affects the whole system the most, so if the Pix2Pix colourization is closer to the original then our system will be closer to VPN.

Overall the results show that a fully automated video colourization model is possible since no coloured frames were used during the whole process. Also, it produces better results than a simple usage of the Pix2Pix software which provides no consistency between each frame, resulting in a bad video output.

6. Related work

In this section we review published work that relates to our project. More specifically we will focus on two papers by (Meyer et al., 2018) and (Xia et al., 2016). The first paper is on Deep Video Colour Propagation and the second is a methodology for robust and automatic video colourization via multiframe reordering refinement.

(Meyer et al., 2018) propose a deep learning framework for colour propagation that combines a local strategy (propagate colours frame-by-frame) and a global strategy (using semantics for colour propagation within a longer range) to colourize a video. Their goal is to "colourize a greyscale image sequence by propagating the given colour of the first frame to the following frames." They take into account the aspects of short range and long range colour propagation. The short range propagation network propagates colours on a frame by frame basis by taking as input two consecutive greyscale frames and estimating a warping function that transfers colours from the first to the second frame. The long range propagation needs semantic understanding of the scene that are extracted by matching deep features extracted from the frames, to transfer colour from the initial coloured frame to the rest frames. The images of these two parallel steps are combined with a convolutional neural network. In their work they compare their results with the methodologies we used in our project, that is Video Colour Propagation (VPC). They suggest that VPC has some limitations and does not yet achieve satisfactory results on video content. While their methodology is presented to be superior to VPC mainly based on PSNR metric comparisons, they do not provide any code for reproducibility of the results. As, a result we couldn't work with their framework in our project.

(Xia et al., 2016) propose a robust, automatic video colourization method that initially estimates motion vectors between a greyscale frame and colour references of it in order to be match using optical flow. Colour is then transferred to matched points in the greyscale image while colour information of matched points is further propagated to other parts of the greyscale image. In addition to that, they designed a multiframe reordering refinement that colourizes sequences robustly.

(Meyer et al., 2018) compare their results with the optical flow and colour propagation methods used by (Xia et al., 2016) and they report that their methodology achieves as good results as the latter but it is considerably less computationally expensive. They also compare their approach to the Video Propagation Network approach of (Jampani et al., 2016), which is the one we used in our project, saying that it has limitations and do not yet achieve satisfactory results on video content.

6.1. Future Work

While in our project we focused exclusively on the implementation of a video colourization system we noticed that the presence of video datasets was scarce. The process of colourizing video is important especially now when considerable effort is given on restoring and colourizing historical video footage. Most notable example is the recent documentary by Peter Jackson for which hundred hours of video footage from the first world war were colourized by hand using advanced computer technology. A procedure like that could take considerably less time if trained colourization models were available. To have that though we need better and more readily available video datasets of various resolutions to mitigate for the immense size that they could take. This way we can allow for more efficient training of such models.

Our colour propagation implementation, while producing satisfying results, could be improved. In fact we referenced works by other authors that showed faster and better performance (Meyer et al., 2018) than our implementation. Implementing this work in the system is a reasonable next step. Another important step is to combine the two models in one system that takes the input video and produces its colourized output immediately. In our project we worked with each part of the system individually so we had to initially import the first greyscale video frame in Pix2Pix and then import its output along with the rest greyscale frames in the Video Propagation Network to receive the final colourized video. While the combination will not have any effect on the final result it will allow for a more seamless implementation of the system.

7. Conclusions

Arguably, the most important learning outcome of our project is that with the lack of a considerably large dataset training of models becomes difficult, even if the models are well designed. This was evident after our first experiment where we trained the model with the Pix2Pix software on the DAVIS dataset. The dataset was proven to be insufficient for successful training of our model so we had to resort in using a pretrained model on the Places365 dataset which was much larger than our initial one.

Our overall research objective was to create a system which takes a whole video sequence in greyscale frames and colourizes them using image colourization and video colour propagation. Our final system was able to perform such an operation with considerable effectiveness on all videos from the DAVIS dataset.

References

Barnston, Anthony G. Correspondence among the correlation, rmse, and heidke forecast verifica-

tion measures; refinement of the heidke score. *Weather and Forecasting*, 7(4):699âĂŞ709, 1992. doi: 10.1175/1520-0434(1992)007<0699:CATCRA>2.0.CO;2. URL https://doi.org/10.1175/1520-0434(1992)007<0699:CATCRA>2.0.CO;2.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Isola, Phillip, Zhu, Jun-Yan, Zhou, Tinghui, and Efros, Alexei A. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

Jampani, Varun, Gadde, Raghudeep, and Gehler, Peter V. Video propagation networks. *CoRR*, abs/1612.05478, 2016. URL http://arxiv.org/abs/1612.05478.

Meyer, Simone, Cornillère, Victor, Djelouah, Abdelaziz, Schroers, Christopher, and Gross, Markus H. Deep video color propagation. *CoRR*, abs/1808.03232, 2018. URL http://arxiv.org/abs/1808.03232.

Nazeri, Kamyar, Ng, Eric, and Ebrahimi, Mehran. Image colorization using generative adversarial networks. In *International Conference on Articulated Motion and Deformable Objects*, pp. 85–94. Springer, 2018.

Perazzi, Federico, Pont-Tuset, Jordi, McWilliams, Brian, Gool, Luc Van, Gross, Markus, and Sorkine-Hornung, Alexander. A benchmark dataset and evaluation methodology for video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Pont-Tuset, Jordi, Perazzi, Federico, Caelles, Sergi, Arbeláez, Pablo, Sorkine-Hornung, Alexander, and Van Gool, Luc. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Xia, Sifeng, Liu, Jiaying, Fang, Yuming, Yang, Wenhan, and Guo, Zongming. Robust and automatic video colorization via multiframe reordering refinement. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pp. 4017–4021. IEEE, 2016.

Zhou, Bolei, Lapedriza, Agata, Khosla, Aditya, Oliva, Aude, and Torralba, Antonio. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40 (6):1452–1464, 2018.