CFNET: A SYNTHESIS FOR VIDEO COLORIZATION

by

Ziyang Tang

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

May 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Baijian Yang, Co-Chair

Department of Computer and Information Technology

Dr. Dmitri A. Gusev, Co-Chair

Department of Computer and Information Technology

Prof. Byung-Cheol Min

Department of Computer and Graphics Technology

Approved by:

Dr. Eric T. Matson

Head of the Graduate Program

To my beloved family for their supports.

ACKNOWLEDGMENTS

I want to gratefully acknowledge my thesis committee for their help and guide. And thanks for the encouragement from my parents in my master careers.

TABLE OF CONTENTS

LIST OF TABLES vii
LIST OF FIGURES viii
LIST OF ABBREVIATIONS ix
ABSTRACT x
CHAPTER 1. INTRODUCTION 1
1.1 Scope 1
1.2 Significance 2
1.3 Research Question 3
1.4 Assumptions 3
1.5 Limitations 4
1.6 Delimitations 4
1.7 Definitions 5
1.8 Summary 6
CHAPTER 2. REVIEW OF LITERATURE 7
2.1 Optic Flow Techniques 8
2.2 Discriminative Model Techniques 12
2.2.1 Colorization 13
2.2.2 Style Transfer 15
2.3 Generative Adversarial Network 16
2.4 Summary 19
CHAPTER 3. FRAMEWORK AND METHODOLOGY 20
3.1 Objective 20
3.1.1 Single Image Generator 20
3.1.2 Optic Flow 22
3.1.3 Generative Adversary Network 23
3.2 Network architecture 24
3.2.1 Image Colorization Generators 24
3.2.2 Flownet 25

3.2.3 GAN Structure 26
3.3 Datasets 27
3.3.1 Kinetics Video Dataset 27
3.4 Training Strategy 28
3.4.1 Adaptive Moment Estimation 28
3.4.2 Batch Normalization 28
3.4.3 Fully Convolution Network structures 29
3.5 Summary 29
CHAPTER 4. EXPERIMENT AND RESULT 30
4.1 Image Colorization 30
4.1.1 Hint Pixel Points 30
4.1.2 Mask Patches 31
4.2 Optic Flow 31
4.3 Inference 31
4.4 Result 32
4.5 Evaluation Criterion 38
4.6 Summary 45
CHAPTER 5. CONCLUSION AND FUTURE WORKS 46
5.1 Conclusion 46
5.2 Future Works 46
REFERENCES 48

LIST OF TABLES

4.1 PSNR between different works 38

LIST OF FIGURES

2.1 Sample of results from flownet2.0 11
2.2 Sample Results from Colorization Models 14
2.3 GAN general idea 16
2.4 DCGAN 18
3.1 Image Colorization Generators 24
3.2 Flownet Structure. K = T-3 in this thesis 25
3.3 Final Network as GAN strucutre. 26
4.1 Traning Losses plot 32
4.2 Results on colorization boats 33
4.3 Results on colorization Monkey and trees 34
4.4 Results on colorization a biker along the walls 35
4.5 Results on colorization a man driving a plane 36
4.6 Results on colorization a plane sliding on the road 37
4.7 PSNR for each classes 39
4.8 Results on colorization of a car drifting on the road. 40
4.9 Results on colorization of a car driving through the dark tunnel. 41
4.10 Results on colorization of a man upside-down 42
4.11 Results on colorization an occluded objects. 43
4.12 Results on colorization an occluded objects with frame-by-frame methods. . 44

LIST OF ABBREVIATIONS

CNN	Convolution Neural Network
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Networks
RGB	Red, Green, Blue color space
L1	L1 Loss Function
CE	Cross Entropy

ABSTRACT

Author: Tang, Ziyang. M.S..
Institution: Purdue University
Degree Received: May 2019
Title: CFNet: A Synthesis for Video Colorization
Major Professor: Baijian Yang

Image to Image translation has been triggered a huge interests among the different topics in deep learning recent years. It provides a mapping function to encode the noisy input images into a high dimensional signal and translate it to the desired output images. The mapping can be one to one, many to one or one to many. Due to the uncertainty from the mapping functions, when extend the methods in video field, the flickering problems emerges. Even a slight change among the frames may bring a obvious change in the output images. In this thesis, we provide a two-stream solution as CFNet for the flickering problems in video colorizations. Compared with the frame-by-frame methods by the previous work, CFNet has a great improvement in allevaiting the flickering problems in video colorizations, especially for the video clips with large objects and still background. Compared with the baseline with frame by frame methods, CFNet improved the PSNR from 27 to 30, which is a great progress.

CHAPTER 1. INTRODUCTION

This chapter provides the introduction to the previous research study in computer vision and deep learning. In the following sections, it will define the research questions, significance and purpose in video colorization problems. It also points out the boundaries in this specific task.

1.1 Scope

Computer vision and deep learning have been applied in different areas in recent years. In 2017, the latest technologies of face recognition have been applied in iPhone X, which arises the wide usage facial related application in mobile phones. Earlier, the object detection has achieved a huge breakthrough using deep convolution neural networks in the ImageNet Challenge. Among the over 1,000 classes and more than 1,000,000 images, the state-of-art algorithm has achieved about 3.47% in top-5 error, which beats the 5% in top-5 error in the performance of human beingsRussakovsky et al. (2015). These inspiring results show the strong abilities of the deep neural network in image feature extraction.

To make use of the powerful feature extraction characteristic in deep neural networks, scientists explore further and apply it to more complicated tasks. One of them is the image-to-image translation problems, which focus on the latent mapping between one input image to the output image. This problem has several specific sub-problems. One of the sub-problem is the style-transferring. Given a styled painting from the world-class artists like Vincent Willem van Gogh, the scientists are trying to transfer the masterpiece style into an arbitrary image. Among these sub-problems, this thesis is exploring the colorization problems. Colorization is similar to styled-transfer but its goal is to generate some realistic images from the input data. Usually, the input data is gray-scale, while the output images are in RGB format. Despite colorization in images, it can also be applied in videos, because videos are the sequence of many similar frames. In general, video

colorization is a more difficult colorization problem in this field. Since it has the temporal consistency and requires the stronger algorithm to make it realistic. All these sub-tasks have the similar idea, and therefore make it easier to transfer one successful solution into another.

Videos translation problems are the extension to image translation problems. Videos are the sequence of images, and they also have the mapping from an input video to an output video. However, videos have the constraint from the context of the frames, which includes the temporal information between previous frames and current frames. This makes the video translation problems more complicated and to our best knowledge, no work has raised a good solution to solve the video translation yet. This thesis mainly focused on the video colorization problem. But similar to the characteristic in image translation problem, the mapping representation of the video colorization problem can also be easily transferred to other video translation problems.

1.2 Significance

Although the deep neural network has achieved huge success in recent years, the drawbacks are also obvious: most of the models rely on supervised learning and need labeled data. And getting those labeled data takes a long time with huge budgets. This thesis provides a self-supervised deep neural network method to train the raw data which is free from the Internet. In today's era, data grows explosively, and images and videos data are easy to access. Many public datasets also help us to make the process of data collection easier. The ImageNet provides more than 1,000,000 images over 1,000 classes. And it also provides a large number of videos. Therefore, the cost is affordable while a large amount of data is guaranteed.

During the past years, scientists have explored a lot in the image-to-image translation problems. However, little work has done in the field of video-to-video translation problems. This brings novelty to this thesis. Besides, the solutions in image-to-image translations problems may not do well in the video-to-video field. However, if one can achieve a great result in video-to-video translation, the algorithm will definitely do well in the image-to-image sides. Therefore, the exploration in the video side is worthy and significant.

1.3 Research Question

In this thesis, we try to find a two-stream method to solve the video colorization problem. This problem can be split into two parts. The first part will colorize the gray scale images, and the second part is to learn the temporal consistency among the sequence of the frames in the video clips.

To solve the colorization problems, we can use the state-of-art framework from image-to-image colorization problems. It will requires the network to learn the spatial information from the input images. After extracting the specific features from the input images, we can use the ground truth color as reference to colorize them.

To learn the temporal consistency, it is actually solving the flickering problems in video-related tasks. Due to the one-to-many mapping property, an input image can have different output images. In this case, a slight change in the input image can bring huge different output. In this case, some prior information from the previous outcome can be useful in choosing the same mapping in the video frames.

In short, the study will first colorize the frames in the videos, and train a model to learn the temporal consistency to make the following outputs similar to the previous ones.

1.4 Assumptions

The assumptions for this study include:

- The mean and standard deviation of the sample dataset can represent for the mean and standard deviation of the population.(i.e. We know that there are millions of video clips from the internet, we assume that the mean and standard deviation of all these video are similar to the mean adn standard deviation in the sample of the training datasets.)
- CNN can estimates the true mapping in finite steps by feeding with a large amount of data.

1.5 Limitations

The limitations of this study include:

- This work only considering the high-end device with at least 6GB Nvidia GPUs.
- The results of the thesis is only compared to the existing image-to-image translation works.
- The results aims at the methods of extending the image colorization problems to videos field, it will not guarantee an improvement of the accuracy from the current image colorization works.
- The video clips should be in similar scenes and not lasts for more than 10 seconds.

1.6 Delimitations

The delimitations for this study include:

- This work doesn't consider the real-time algorithms.
- This work only show the possibilities that deep networks can do well in video-to-video colorization. It will not guarantee that it can do well in any gray-scale videos.

1.7 Definitions

In the broader context of this thesis, we have the following terms:

Convolution Neural Network(CNN) - CNN is a class of deep, feed-forward artificial neural networks. It used a non-linear operation call convolution to extract the most important feature from the images. It can also stack the feature together using different kernels. To save the number of parameters, the kernels will share the same weight, which also lead to the property of shift invariant.(i.e. the feature of an object can still be extracted even after translation or rotation.)(LeCun et al., 1995)

Loss Function - The prediction output can be different from the ground truth. In supervised learning, the goal is to minimize the different from the predictions and the ground truth. And the function to minimize the different is called loss function. Usually, loss function is simple for derivation and calculating the gradients.(Wald, 1950)

Generative Adversarial Networks(GAN) - GAN is a class of machine learning algorithms used in unsupervised machine learning or supervised learning. It has a generator and a discriminator, which contest with each other in the training phase. The generator tries to fool the discriminator by generate realistic images, while the discriminator tries to tell the difference between real and fake.(Goodfellow et al., 2014)

RGB color space - The RGB color space is combined with the three channel of Red, Blue and Green. With the three primary color, it can reproduce other colors by linear combination from the three main colors.(Pascale, 2003)

CIELab color space - the CIELab color space is a color space defined by the international Commission on illumination(CIE) in 1976. It separate the Lightness from the color pairs. The L for the lightness, while a and b channel for the green-red, blue-yellow color components. Compared with RGB colro space, CIELab is more suitable to human color vision, which has the same numerical change in the values that forms the color in human visions.(Robertson, 1977)

1.8 Summary

This chapter introduced the research questions in this thesis, pointing out the significance, scope and limitations. Additionally, this chapter provides some definitions that being used in the rest of the research.

CHAPTER 2. REVIEW OF LITERATURE

Video colorization is a specific problem in image-to-image translation problems.To solve it, a review of the the previous work provides a good start. However, the solutions of image-to-image translation problems cannot be directly transferred to video-to-video problems. Videos contain temporal information, which results in temporal consistency between frames. To be more specific, the frame should look similar to the frames that next to it. For example, if the current frame has a blue car, in the next frame, the car cannot disappear suddenly, and the color should not be changed. However, in the solution from the image-to-image translation is directly applied in video areas, the temporal consistency is hard to be maintained because a slight change of the input images may change a lot in the output. This is also be called as flickering problem. Therefore, in solving flickering problems, the model must be able to remember the information in the previous frames. To achieve this goal, optic flow has been proved from other similar works in video object detection and video style-transfer.

To solve the image-to-image translation problems, there are two different models: discriminative models and generative models. Discriminative models are probabilities problems. The idea is that given a condition X, the model determines the possibility of variable Y based on the condition X. A general formula of a discriminative model is $D(X,Y) = P(Y|X)$. Since each pixel has the range from 0 to 255, discriminative models can determine the possibility of the pixel value with the given labels. The advantages of discriminative models are that they have a few parameters, and can produce the result fast. However, the limitations are also obvious. The models are limited by the given condition X, which are the labels of the images. This results in week generalization abilities. When testing the model with some images which contain some objects that never appear in the training dataset, the model provides bad results. To enhance the robustness of the models, researchers usually enlarge the training dataset. However, even the largest existing dataset cannot contain all the objects in the world. Therefore, the limitations still remain unsolved yet.

Dislike the discriminative models, generative models first set up a model to map from X to Y. That is, given the input of X, generative models can ”generate” the output of Y. The general formula of a generative model is $Y = G(X)$. In the field of image-to-image translation problems, a generative model can make up a new image from the input image. Generative models can solve the limitations of discriminative models. Because the models generate the output images from the given images so that the output images will not be affected even the input images do not appear from the training dataset. However, since the output images are generated by the computers, they can not be very photo-realistic. But if the output images are not photo-realistic, the models are actually useless. So the limitation of the generative models is that we cannot find a generative model with reasonable results easily. Nowadays, the most reliable generative models are the generative adversary networks(GAN). In GAN, it first trained a generative network to get the output images, then use a discriminative network to differentiate the output images from the real images. If the output from a generative network can fool the discriminative network, then the output will be much more reliable.

2.1 Optic Flow Techniques

Optical flow is not a novel idea that coming out in recent years. It was first pointed in 1950, Gibson (1950) defined that optical flow as a pattern of the movement of objects in the scenes. There were two methods to get optic flow. One was using traditional computer vision methods, which was non-learning based methods, as the following description:

1. based on global and local image patches and features. One of the typical algorithms is Lucas-and-Kanade.Lucas, Kanade, et al. (1981). In Lucas-Kanade Method, it assumes the velocity $v = (v_x, v_y)$ to be constant over a small neighborhood Ω_x of every $x \in \Omega$.

2. based on image sequences. This is also been called Frequency-based methods. The method is usually applied to videos fields to calculate the relative moving distance of a pixel from one frame to another frame which is next to the previous frame.Humphreys and Bruce (1989)

3. based on Region matching. In this kind of method, it first defines the velocity v as the shift $d = (d_x, d_y)$. Then it calculates the sum-of-squared-difference between two frames for the optical flow.

Optical Flow cannot be applied in any scenes, it still has some limitations. To apply it, the image sequences must match the following requirements:

1. Brightness constancy. The optical flow requires a constant brightness of the image sequences. The constant doesn't mean that the brightness cannot change at all. A slight change is allowed but a sudden great change of the brightness will hugely damage the prediction of the pixel movement of the next frame because the calculation will raise a huge difference from the accurate result.

2. Small motion. The optical flow also requires the movement of a pixel from one frame to another frame cannot be to fast. For example, if the pixel was located on the leftmost side of the $frame_1$, then it moved to the rightmost side in $frame_2$, even the best optical flow estimation algorithm nowadays will fail in tracking the pixel because it moves too and the difference between the prediction and ground truth will be too large.

In exploring the video-to-video translation problems, scientists found that optical flow is suitable to track the pixels in videos. The videos are the sequences of frames. In two adjacent frames, the brightness can be considered as constant in most cases. And the pixel movement is small because a video usually has more than 30 frames in a second, and even the fastest object in the reality will just move slighted in $1/30$ seconds.

Another method is learnig-based optic flow methods. We know that the traditional methods must have a complicated function for all different datasets. However, this general methods may have bad results for a specific dataset. In this case, with the rapid development in deep learning, which can be considered as a powerful tool in fit a

complicated function in a specific dataset, reserachers designed the learning-based optic flow methods, using CNNs to estimate the pixel movements among the frames in the video clips. In 2015, Dosovitskiy et al. (2015) published the first Flownet using learning based methods. The author designed two architectures called ”FlowNetSample” and ”FlowNetError”, which can calculate the optical flow using convolution neural networks. Besides, the author also made a dataset called ”Flying chairs”, which can automatically generate labels for the optic flow from the image sequences. By training the ”FlowNet” using the dataset, the convolution neural network model can be applied to other datasets and achieved state-of-art results at that time.

But ”FlowNet” still has limitations. Because it was the first work in applying optical flow into deep learning, its state-of-art result was only compared with the traditional algorithms. And the dataset of ”Flying chairs” only revealed a small part of the real dataset. In 2017,Ilg et al. (2017) updated it into ”FlowNet 2.0”. In ”FLowNet 2.0”, the authors advanced the concept of end-to-end learning of optical flow. They made three major improvements:

1. They focused on the training data and provided an optimal schedule in training the data.

2. They develop a stacked architecture to increase the accuracy of the networks.

3. They used a special designed network to deal with the small displacement in the videos.

According to the results from the paper, ”FlowNet 2.0” has the similar speed to ”FlowNet”, which can achieve at most 140 fps, but it decreases the error by more than 50%.

Others have applied the work of ”FlowNet” and ”FlowNet 2.0” into some specific computer vision problems. X. Zhu, Xiong, Dai, Yuan, and Wei (2017) successfully apply the optical flow into video recognition problems. In the paper, the authors used ”FlowNet” to learning the moving distance of each pixel of the current frames, then they generate the

next frame by warping the current frame with the result of optical flow, which is the result from the ”FlowNet”. In this case, instead of detecting all frames in the videos, the authors can only detect half of the frames and warping the other frames using optical flows. The method speeds up the process of video recognition without decreasing the accuracy.

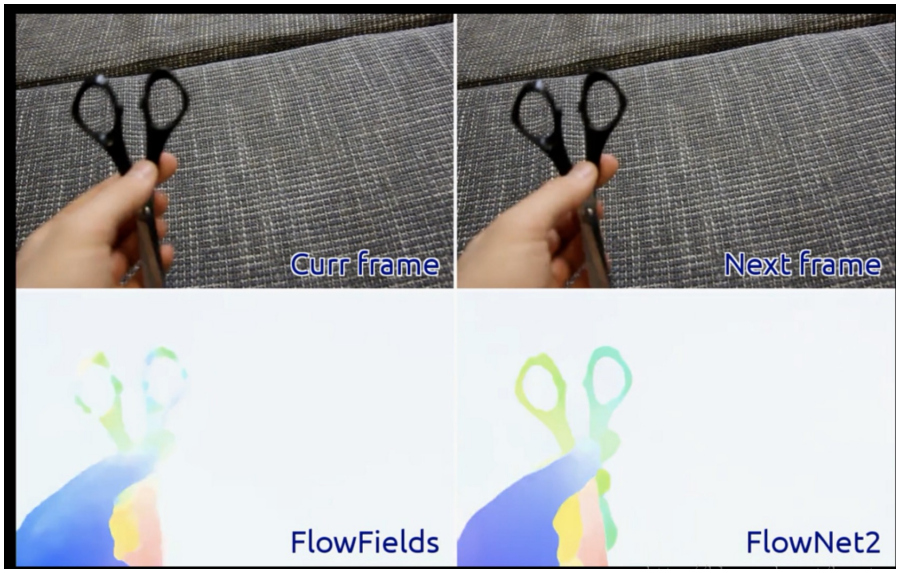


Figure 2.1. Sample of results from flownet2.0

Some researches (X. Zhu, Wang, Dai, Yuan, & Wei, 2017) also stated another application using optic flow. In this paper, the author presented a flow-guided method using deep networks, which embeded the process of fitting a flow mapping function in the network and provide an end-to-end framework for video object detection. Despite the work in ”Deep Feature Flow” mentioned above, the author only computed the expensive convolution on the keyframes and using the optical flow to warp the rest frames. They also showed that the method using optical flow can not only speed up the computation but also track the fast moving objects better.

From the publications above, optical flow can be considered as a mature and reliable method in memorizing the previous frames and warping the future frames at relatively fast speed. In this case, it provides an idea in using warping the frames from the current frame in the video then apply the image-to-image solutions to each frame. Since

the stream of compute the optic flow is different and independent from the other stream of applying the image-to-image solutions, the methods can be conducted as two-stream CNNs.And due to the property of optic flow in keeping the temporal consistency, the two stream methods has been used widely in extending the solutions in images to video fields.

2.2 Discriminative Model Techniques

Discriminiative Models are a direction in the image-to-image translation tasks. Discriminative model will train a model to fit a mapping function to choose the most likely results from the given probabilities. Among the image-to-image translation problems, the discriminative models have been used in many tasks, especially in Super-pixel, colorization and style transfer. In all these tasks, researches can provide the potential outputs and let the model to choose the most likely output among them. Among all the tasks, colorization and style transfers share many related features, and therefore, in the following literature reviews, we will mainly focus on the previous work in colorization and style transfer.

1. Super-pixel has the input can be low-resolution images, and generate an output image with high-resolution. In Super-pixel techniques, the discriminative model can enlarge the low-resolution images and generate detailed features in high-resolution images.(Stutz, Hermans, & Leibe, 2018)

2. Style transfer is another sub-research problem in image-to-image translation tasks. The input can be more than one image. For example, there can be 2 input images. One is an original image and the other can be a styled painting (i.e. a masterpiece from a famous artist). After the translation, the output is the original image with the style from the painting. (Luan, Paris, Shechtman, & Bala, 2017)

3. In colorization, the input is a gray-scale image and the output images colorize the input. In the past, people use the label to supervise the values of the three RGB channels. In this case, the output colored images will have the similar color with the labels. Besides, previous work also make use of other color space like CIElab. This

is because the grayscale image represents the lightness of the image, and we can keep the lightness information to maintain the spatial consistency of the images and only learn two ab channels, which represents the green-red and blue-yellow components. Iizuka, Simo-Serra, and Ishikawa (2016)However, the color of the object can be different. For example, the apple can be red, and it also can be green. But in using discriminative models, the output tends to compute the color with the possibilities. So the possibilities of the red may be larger than the possibilities of the green and output images will color the apple into the red.

2.2.1 Colorization

In the research of video-to-video translation, the input images are the frames from gray-scale images with the segmentation of the objects. And the output images are the colorful images in an RGB format with photo-realistic scenes. In this case, the translation a single frame into the desired output, the most relevant techniques are the colorizing and style transfer. In colorizing, the coloring process is like generate the value of R, G, B channels from the gray-scale value for every pixel in the input images. In this case, the output of can be the most likely probability of the value ranging from 0 to 255. In the videos, given a gray scale frame with the label of segmentation, the output image is an RGB frame with the segmentation. The segmentation is an outlined region that cropped from the images to show the position of a specific object. Given this condition X of the segmentation, the whole task is a problem to determine the most possible value of the colors for each pixel in the frames.

There are two different ways in colorization. One is to colorize the gray-scale images with the label of colors. In this case, if the label of the apple is red, the model will prefer red to green in generating the output images. Current state-of-art of this method is the work from Iizuka et al. (2016). They combined the both the global and the local features of the images. Besides, they trained their model with a dataset containing more than 1,000,000 images. As I have mentioned above, the large dataset can reveal the problem in discriminative models that overfitting to the biased label of a specific color. In

other words, the large dataset provides red apples and green apples as training color, and therefore the model can learn two different colors of the apples and may provide more robust results. Another idea in colorization is that providing vibrant and realistic colorizations, regardless whether the color of the output is similar to the label of the color. Zhang, Isola, and Efros (2016) and Larsson, Maire, and Shakhnarovich (2016) are two works in colorization that generating realistic colored images. They are both end-to-end convolution neural network discriminative models. Despite the comparison of the output images and ground truth images, Zhang et al. (2016) evaluate the results by asking some human to judge whether the image is faked or not. According to their result, 32% of the faked images fooled human eyes, which is significantly better than the previous works. To achieve this, the author designed a loss function to re-weight some rare color and provided a special method called ”annealed-mean” to allocate the final color to the output images. Besides, the author also provides a huge dataset to reveal the limitations of the discriminative models.

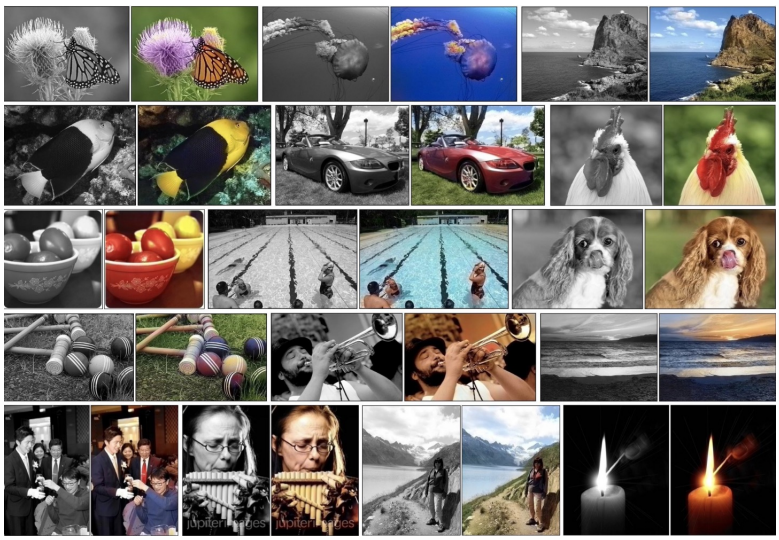


Figure 2.2. Sample Results from Colorization Models

Larsson et al. (2016) at the meantime, developed a model to predict the per-pixel color histogram, and using the histogram as a prior condition for the color formation.

In addition, evaluating the generated colorized images is also difficult. One straight-forward way is generating the output images, and ask person to judge whether it is faked or real. If the output images can fool human eyes, it should be considered as photo-realistic. Nowadays, there is a platform called "Amazon mechanical turk"(AMT), that allow people to upload image into the platform and have judges to evaluate the results. The state-of-art of the image colorization have 32% output images that can fool human eyes, while the ground truth have the accuracy of 50%. The ideal accuracy is 50% because the people can only guess whether the image is real or not.

2.2.2 Style Transfer

In styled transfer, researchers have found a way to transfer the styles into a video. Therefore, reviewing the previous work in styled transfer can provide us not only the ideas of colorizing the images, but also some tricks in dealing with the videos. Chen, Liao, Yuan, Yu, and Hua (2017) stated an end-to-end online video transfer methods that can consider the short-term temporal information to make sure the consistency in video style transfer. If they the ensure the consistency in short-term frames, it can provide long-term consistency because F_n is consistent to F_{n-1} and F_{n-1} is consistent to F_{n-2} ..., and F_2 is consistent to F_1. Therefore, through the transmission from F_n, we can easily prove that all the frames are temporal consistent. To achieve this, the author designed an architecture using Encoder and Decoder to trained a network to extract the difference of the positions from the same feature. They also added a mask network to deal with the condition that some objects in the video disappears in the middle of the sequence and re-appear in the future frames. Finally, they showed that the trained sub neural network can be transferred to a brand new painting style without extra training. This indicates the robustness of the sub-network.

2.3 Generative Adversarial Network

Generative models demonstrate another direction in solving the image-to-image translations. It generates a model to set up the mapping from each pixel in the input images to output pixel. Nowadays, the most efficient generative models are called generative adversarial network(GAN).(Goodfellow et al., 2014). As the following figure shows, the author designed a generator to generate the output images and trained a discriminator to differentiate the output images and ground truth. The training process is adversarial. The generator is trying to generate images as realistic as possible. At the

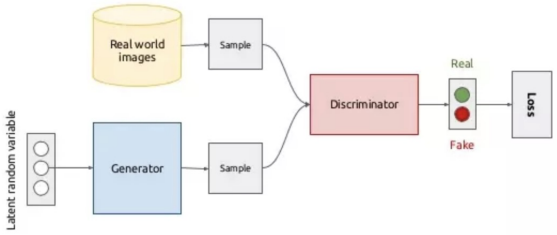


Figure 2.3. GAN general idea

mean while, the discriminator will try to differentiate the predicted images from the ground truth. When the training is done and everything goes well, the generator should generate fake images that can fool the discriminator. Statically, the loss of the generator will be low while the confidence of the discriminator will be close to 0.5. The confidence is close to 0.5 because the discriminator only has about 50% confidence to tell whether the image is faked or not. And the 50% is just the chance of guessing. In this case, the discriminator will fail to function properly. Even though low confidence like 0.2 means that the discriminator only has 20% of accuracy to label the real images, it doesn't work in GAN because taking the opposite prediction will lead to high accuracy. That is, when the confidence is low, we can label the faked images predicted by the discriminator as real images, and label the real images predicted by the discriminator as faked images. In this case, we have 80% confidence that the output images are labeled correctly. Therefore, the results will still be not realistic at all.

Although the original GAN can lead to great generative models, it has many limitations. First of all, the training process must be very careful. A slight change in the super-parameter in the neural network will cause a huge difference in the output results. So the training model must be designed carefully, and each step in the training counts. Besides, there were no constraints in the training models, which results in unexpected output in the generated images. This is also called unstable generative models. For instance, the shape of apples is similar to the shape of the oranges, because they are both sphere objects. Since the CNN is a black box, which researches cannot explain why it works in theory. And researchers can only control the input images, the output from the original GAN may generate an orange while the input is an image of an apple. As the number of classes for classifiers grows, the chance of unexpected results significantly grows. Consequently, GAN is hard to be trained and the good generators are not easy to get.

To solve the limitations mentioned above, Radford, Metz, and Chintala (2015) did some changes to the original architecture in GAN and brought a relatively stable model, which is called as DCGAN. The main contribution of DCGAN is applying vector arithmetic into the images. In deep learning, the input images will go through several filters in each convolution layers, and each filter can extract one feature from the input feature maps. The space of the combination of these features is called the latent space. The latent space usually has very high dimensions, e.g. $2048 * H * W$, and the hidden spaces are connected by a large number of weights. Each of the weights is represented by a vector. The author of DCGAN states that the original GAN is not stable because it only has one vector to represent a feature. After some experiments, the author found that the GAN can be much more stable is a single feature can be represented by some arithmetic computations. As the following figure showed, the smiling man can be generated by the features of a smiling woman minus neutral woman and add neutral man. By the techniques of vector arithmetic, the results of DCGAN are a linear combination of the features in the hidden space. Therefore, the output images can be much more stable.

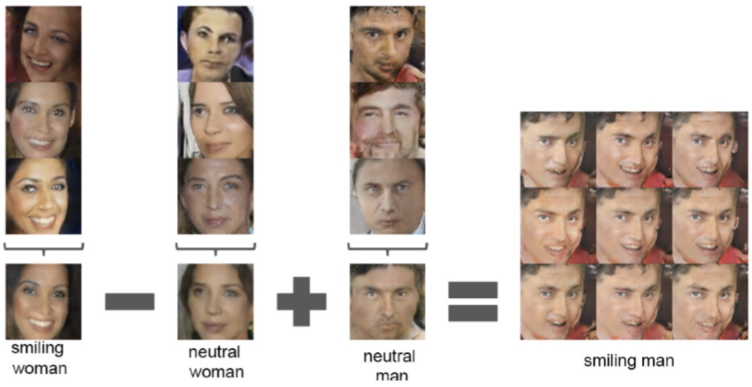


Figure 2.4. DCGAN

Although DCGAN somehow solve the hard training problem, it cannot be directly applied into the image-to-image translation tasks because the input of the generator is an one-dimensional vector as a Gaussian noise to the real input image. The following works realize the problem and solve it by providing a U-net structure and replace the random Gaussian noise with the input image. Isola, Zhu, Zhou, and Efros (2017) generalize the problem and state a general image-to-image translation model called as ”pix2pix”. In this work, the input and output images form a pair, and the author take the input image as the input of the generator, and generate an output image to compare with the output image in the discriminator. They also find that the same structure can be transferred to several similar tasks in image-to-image translation tasks with different dataset. From the results in their papers, the same structure can map the ”labels to street scene”, ”labels to facade”, ”colorization”, ”aerial to map”, ”day to night” and ”edges to photo”. Then the same team raises a follow-up work called ”cycle-gan” (J.-Y. Zhu, Park, Isola, & Efros, 2017) to solve the paired input-output dataset. That is, to train a model with the ability of mapping from one edge to a photo, the input edge can use more than a photo as the output image, while the output image stays the same in pix2pix.

In video related problems, the idea of GAN seems a good idea. Considering a large number of objects may appear in the videos, the largest current dataset cannot cover all the classes of the objects. But the generative models can produce the objects by doing vector arithmetic computations. Wang et al. (2018) states a similar video-to-video

synthesis using GAN. The paper points out a sequence generator to predict the next frame in the videos. Then use the ground truth of the next frame to discriminate the generated frame, and train the network. The input is a gray-scale segmentation images sequence and the output is an RGB photo-realistic scene video. According to the results from the paper, when swapping the value of trees and buildings in the input,(the input is segmented when the pixel with value 1 represents tree and the pixel with value 2 represents building, the swap means the pixel valued 1 represents building and pixel value 2 represents tree), the output also swap the position of trees and buildings.

2.4 Summary

This chapter summarized the related previous works and introduces the usage of optic flow as a tool in exploring video data. It also states some attempt in recent years how to combine two ideas in different areas and generate great results in deep learning and computer vision. And it also demonstrates some previous works that can be compared in the experiment.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

This chapter will introduce some ideas used in the previous work and demonstrate how this thesis combine some of the ideas into an end-to-end network. To colorize a video, the framework needs a generator to colorize each frame of the videos. To learn the correlation between the frames, this thesis adopted a learned-based optic flow method. In the training phase, the thesis also used two discriminators to help the CNN to learn the colors better.

3.1 Objective

The following will discuss my method, CNN framework and derive the objective loss functions for the whole network.

3.1.1 Single Image Generator

Previous work has shown many results in single image colorization. This thesis made use of those ideas and trained a colorization generator with a 10 layers U-net structure. The first seven CNN layers down-sampling the input data to extract features. And the last three layers worked as up-sampling and retained the output size as the input data. In this generator, the CNN learned to minimize the predicted color with the ground-truth color.

$$L_{reg} = L_1(Y_{gt}, Y_{pred}) \tag{3.1}$$

However, previous work showed that directly apply this objective loss function will lead to blur and grayish problems. This is because the L1 loss will learn the colorize with minimal distance of the mean of the color pairs. And the mean of the color pairs leads to grayish results. To partially avoid this averaging effect, we used the one-hot

encoding to treat the color pairs as individual classes. Hence the colorization task can be treated as a classification problem, and we have the classification loss:

$$L_{cls} = L_{CE}(Pred, class) \tag{3.2}$$

in this equation, the *class* represents the probability of the output in that specific class, and j is an index to sum up the probability in all class.

To implement to one-hot coding on the color pairs, we separate the color from the lightness in the images. We used the Lab color space to get the color ab pairs, with the following process:

convert RGB to sRGB

$$v \in \{sR, sG, sB\} \tag{3.3}$$

$$V \in \{R, G, B\} \tag{3.4}$$

$$v = \begin{cases} V/12.92 & \text{if V} \leq 0.04045 \\ (V + 0.055/1.055)^{2.4} & \text{otherwise} \end{cases} \tag{3.5}$$

convert sRGB to CIE xyz

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3675 & 0.1804 \\ 0.2126 & 0.7151 & 0.0721 \\ 0.0193 & 0.1192 & 0.9603 \end{bmatrix} \begin{bmatrix} sR \\ sG \\ sB \end{bmatrix} \tag{3.6}$$

convert CIE xyz to CIE Lab

$$\begin{cases} xs = X/0.95047 \\ ys = Y/1.0 \\ zs = Z/1.08883 \end{cases} \tag{3.7}$$

$$xyz = \{xs, ys, zs\} \tag{3.8}$$

$$xyz = \begin{cases} 7.787xyz + 0.1379 & \text{if xyz} \leq 0.008856 \\ xyz^{\frac{1}{3}} & \text{otherwise} \end{cases} \tag{3.9}$$

$$\begin{cases} L = 116y - 16 \\ a = 500(x - y) \\ b = 200(y - z) \end{cases} \tag{3.10}$$

We went through the whole dataset and calculated that all the color pairs were in the range from $[-110, 110]$. In this case, we divided the range into 23 bins by divided it by 10. Then we will have (23 * 23)=529 classes for all the color pairs. In training, we regard the colorization tasks as a 529 classes classification problem.

Finally, we can derive the objective loss function for single colorization generator:

$$L_{cG} = L_{cls} + L_{reg} \tag{3.11}$$

3.1.2 Optic Flow

We implemented a learning-based optic flow method to memorize the temporal coherence in the videos. Our goal is to alleviate the flickering problems in video colorization problems. In the implementation, set L =$\{L_1, L_2, ..., L_n\}$ is the input sequence. And set C=$\{C_1, C_2, ..., C_n\}$ is the real color pairs of the sequence. $\mathscr{F}$ is the optic Flows,

which is the function we want CNN to learn. $\mathscr{W}$ warps the flows into the previous frames. Ideally, we should minimize:

$$L_{flow} = L_1(\mathscr{W}(C_i, \mathscr{F}(L_i, L_{i+1})), C_{i+1}) \tag{3.12}$$

But in the inference period, we do not have real color pairs, so we should use the generated colors from previous frames instead. Let set fC = $\{fC_1, fC_2, ...fC_n\}$ to be the generated color pairs from colorization generator. We add an additional loss:

$$img_{warp} = \mathscr{W}(fC_i, \mathscr{F}(L_i, L_{i+1})) \tag{3.13}$$

$$L_{warp} = L_1(img_{warp}, C) \tag{3.14}$$

In some cases, when the video clips were long, some object might be occluded. In this case, we used M as a mask to determine whether the current pixel should use the color from warped images or colorization results.

$$img_{mix} = M \odot img_{warp} + (1 - M) \odot fC \tag{3.15}$$

$$loss_{mask} = L_1(img_{mix}, C) \tag{3.16}$$

finally, we have the objective loss function of flow:

$$L_F = loss_{flow} + loss_{warp} + loss_{mask} \tag{3.17}$$

3.1.3 Generative Adversary Network

The objective loss function of a conditional GAN is like the following:

$$L_{GAN}(G, D) = E[log(D(y))] + E[log(1 - D(G(x))] \tag{3.18}$$

From previous work, adversarial training can generate images sharper. Therefore, in this thesis, we add a discriminator and give some hints to our colorization generator.

From the above, we have our full objective functions:

$$Loss = L_{cG} + L_F + \lambda L_{GAN} \tag{3.19}$$

In the implementation, the discriminators was implemented with small networks. In this case, to enlarge the affect of the discriminators, we give more penalty to the Loss in GAN. In this study, the $\lambda = 10$.

3.2 Network architecture

3.2.1 Image Colorization Generators

The encoder and decoder will adopt a VGG based U-net structure as the following figure shows:

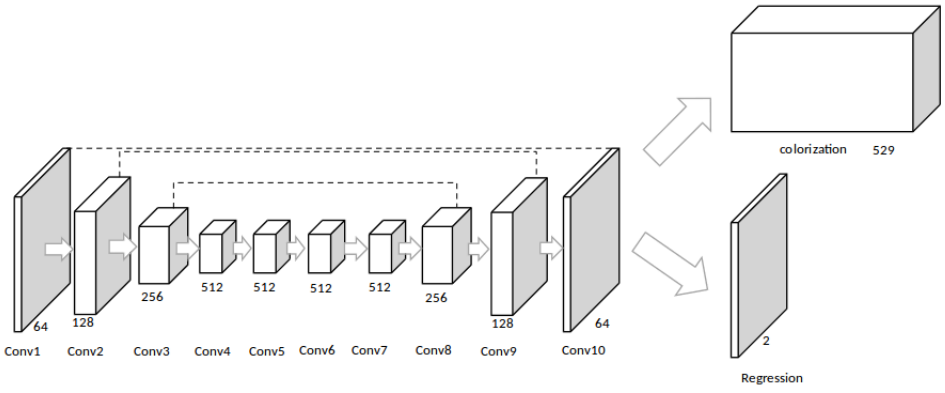


Figure 3.1. Image Colorization Generators

In this network, we designed a 10 layers CNN as a shape of U-net. For the first three layers on the left sides, served as downsampling and extracted the global features from the input. In the middle four layers from Conv4 to Conv7, it worked as a bottleneck, and the last three layers used transposed convolution layers for upsampling. And a skip-layer mechanism was used to concat the features from higher layers with the details from deeper layers. All these 10 layers will share the weights. Finally, we developed two different methods to produce the ab pairs. We used a classifier to the most likely class from the 529 encoded classes. Besides, we also implemented a regression method to the regression results. As the above describes, we combined the two losses as the generator.

The input image is a gray-scale image with a size of $1 * 256 * 256$. To enhance the colorization results, we also add a hint of color pixels and randomly masked some patches from the input gray color. The hint color was generated randomly from the 529 classes. And the mask will block a patch of the gray input image. The hint color pixels had a size of $2 * 256 * 256$, and the mask had the size of $1 * 256 * 256$. Therefore the total input size for the Image colorization generator was $4 * 256 * 256$.

3.2.2 Flownet

To maintain the temporal consistency from the videos, this thesis adopts a two-stream method using the learning based flownet structures.

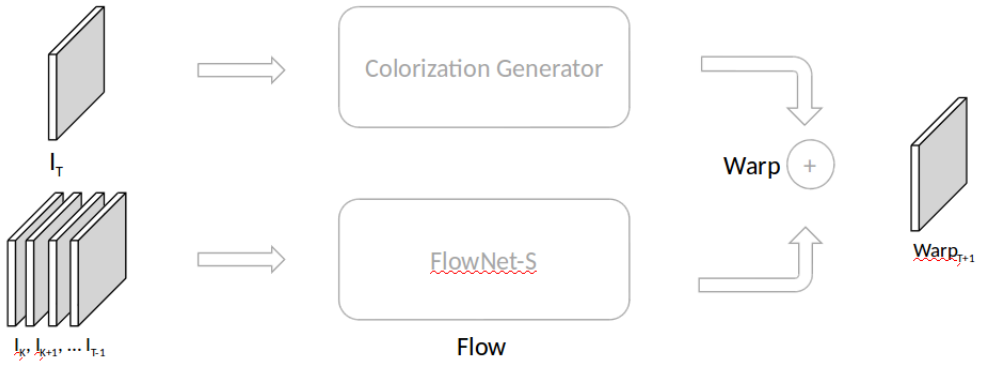


Figure 3.2. Flownet Structure. K = T-3 in this thesis

The input of the flownet will be a set of previous frames $I_K, I_{k+1}, ..., I_{T-1}$ (K=T-3 in this thesis), and the flownet will generate an optic flow f_T as an output. Meanwhile, there will be a result g_T from I_T, and the warp will mix the result from g_T and f_T by a hyperparameter *weight* learned from the network.

3.2.3 GAN Structure

We also designed a GAN structure to make the output images as sharp as possible. The final network structure will like the following:

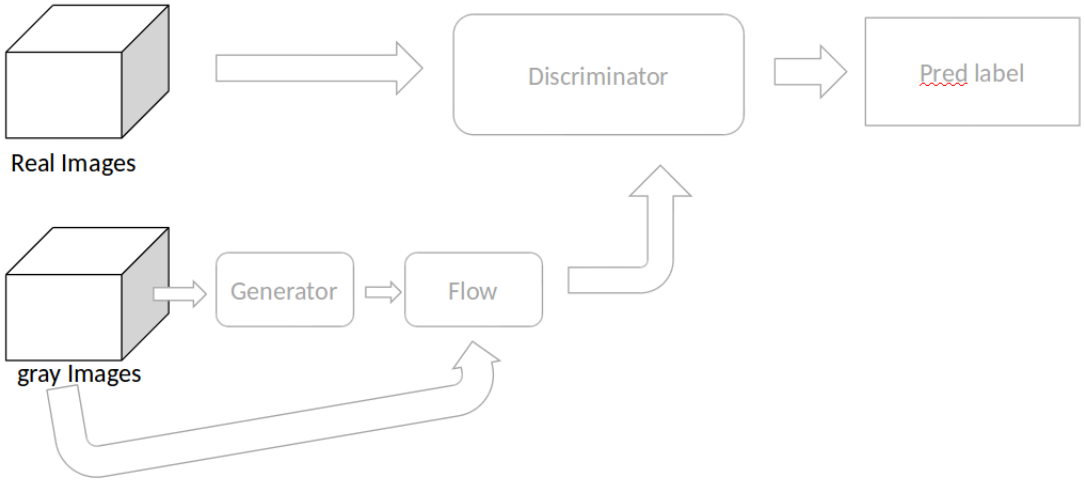


Figure 3.3. Final Network as GAN strucutre.

The generator has been introduced in section 3.2.1, and the Flow structure has been described in section 3.2.2. And the final warped image will be judged from a discriminator.

3.3 Datasets

Davis Video Dataset Davis(Densely Annotated VIdeo Segmentation) Khoreva, Rohrbach, and Schiele (2018) contains about 90 videos with more than 15,000 frames of images. The dataset covered different scenes from outdoor. The dataset was not large, but it was great to test the performance of the models.

ImageNet Video Dataset ImageNet video datasetRussakovsky et al. (2015) is oriented for the task of video segmentation. Fortunately, the labels of colorization tasks are easy to access because all the frames in the videos are color images. To generate the data and label, all we need to do is converting the colorful images into gray-scale, and the gray-scale images will be the data while the original colorful images will be the ground truth. In this case, to train a good model in video colorization problems, a large colorful video dataset is all we need. In ImageNet VID, it provides a 100GB dataset with 30 objects. This is a great dataset to train video colorization models among these 30 objects.

3.3.1 Kinetics Video Dataset

Kinetics Video DatasetKay et al. (2017) has more than 300,000 videos from YouTube. Compared with the dataset from ImageNet VID, the videos have more objects. This is great to refine the model we trained from the ImageNet dataset and enhanced the robustness of our model. Therefore, it provides a large number of objects in different shapes. As mentioned in chapter 2.2 and chapter 2.3, the deep networks need plenty of data to learning some strong representation from the images and videos. In the image colorization, the researchers have used over 1.3 million images to train a great model. The number of data should be larger in training a good model in videos because videos contain much more information than a single image.

3.4 Training Strategy

3.4.1 Adaptive Moment Estimation

For the training phase, this thesis will use adaptive moment estimation(Adam).(Kingma & Ba, 2014) optimization and weight initialization. Adam is a method used for stochastic optimization and it has been widely used in recent years. It is adaptive to both step size and momentum in gradient decent. It has two hyper-parameters in the machine learning models:

$$\begin{aligned} m_t &= \eta[\beta_1 m_{t-1} + (1-\beta_1) g_t] \\ v_t &= \beta_2 + (1-\beta_2) {g_t}^2 \\ \theta_{t+1} &= \theta_t - \eta \frac{m_t}{1-\sqrt{v_t}+\varepsilon} \end{aligned} \tag{3.20}$$

In the formula, the η is the learning rate, β_1 and β_2 are two hyper parameters for updating the gradients and momentum. The ε is a very small number to prevent the equation from divided by zero. In weight initialization, the η is set as 0.001, while $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1e-8$.

3.4.2 Batch Normalization

In training GANs, one of the main concerns is that the generators emit some outputs after going too deep from the convolution neural networks. The phenomenon is called mode-collapse, also known as the Helvetical scenario.Goodfellow (2016) When the mode-collapse occurs, the generators can find a way to consistently fool the discriminator. However, this is not what we want. Instead, the generator should learn strong

representations from the input rather than just find a short-cut to fool the discriminator. Batch normalization has been proven as an efficient way to train both the networks and prevent the generator from collapsing all samples to a single point. The batch normalization layers will be applied behind every activation function layers.

3.4.3 Fully Convolution Network structures

FCN has no fully connected layers. Removing the fully connected layers can significantly decrease the number of parameters in the models, which requires less memory in the GPUs. Besides, the fully connected layers will not keep the spatial information of the input images, but the task of colorization should have the same size between the input images and output images. In this case, fully connected layers are not suitable for this task. Finally, all convolution neural networks allow a flexible input size for a given image, but fully connected layers will fix the input size to match the matrix multiplication. For the colorization problems, the input images will have different input size, and therefore the re-scale processing will somehow impair the data in the beginning phase.

3.5 Summary

This chapter provided a mythology that the thesis used, and conduct the object functions and provide CNNs. It also gave the training strategy and demonstrated the datasets.

CHAPTER 4. EXPERIMENT AND RESULT

In the training phase, due to the limitation of GPUs, we trained the model separately beforehand and used the model we had as the pretrain model and trained the whole network together in the end. When training the whole model, we set the batch size as 2 (two videos) and epoch iteration we trained a contiguous three frames at the same time. We trained five epochs, with an Adam optimizer started with a learning rate of 0.001 and reduced to 0.0001 in the next two epochs and 0.00001 for the last two epochs. The training time depended on the size of the dataset and the memory of the GPUs. In our experiment, it took about a week to train one epoch with the dataset of ImageNet with two GPU of 8GB Nvidia 1070. And it took about 12 hours to train one epoch with Davis 2017 with the same GPUs.

4.1 Image Colorization

In the phase of training singe image generator, we applied hint pixels points and one mask patch in the input image. The hint pixel points were generated in a random position of the image, and the value of the color came from the 529 classes from the one-hot encoding classes. And we also block a 10*10 image patch randomly for data augmentation.

4.1.1 Hint Pixel Points

We set a probability as $p = 0.125$ to add hint pixel points. When applying these pixel points, we chose 10 pixels from the positions randomly generated from the normal distribution. And we randomly chose the color from the encoded 529 color classes for each point. The goal for hint pixel points was to encourage more color selections from the same input image because each object might have different colors.

4.1.2 Mask Patches

We set a mask patch with the size of 10*10 and set the value of the pixels as 0. The mask patch was used for data augmentation in the CNN, and enhance the generalization in the model while preventing overfitting.

4.2 Optic Flow

In the dataset, we don't have the ground truth of the flow. Therefore, we cannot directly minimize the objective function from the predicted flow with the real optic flow. Fortunately, we can get flow results from the state-of-art results. In this thesis, we made use of the results from the flownet2 as the ground truth of the flows and pretrained our flownet model based on that ground truth. Besides, we also have two inputs $frame_{prev}$ and $frame_{post}$, and we compared the warped image from $frame_{prev}$ and predicted flow with the $frame_{post}$. And our goal was to pretrained the mask weights by minimizing the wrapped result to the ground truth. .

In end-to-end training, instead of using $frame_{post}$, we used the faked color pair generated from the Image colorization generator. We trained like this because in the inference, we didn't have the ground truth color pairs, and therefore we had to use the faked color pairs instead. Therefore, in training, we also used the faked color pairs as input and used the ground truth color pair to minimize the loss between the fake color with the real color.

4.3 Inference

In the inference, our input will be a sequence of gray images. We colorize the first frame and calculate the flows and get the rest colorized frames by warped the flow and previous colorized frame.

4.4 Result

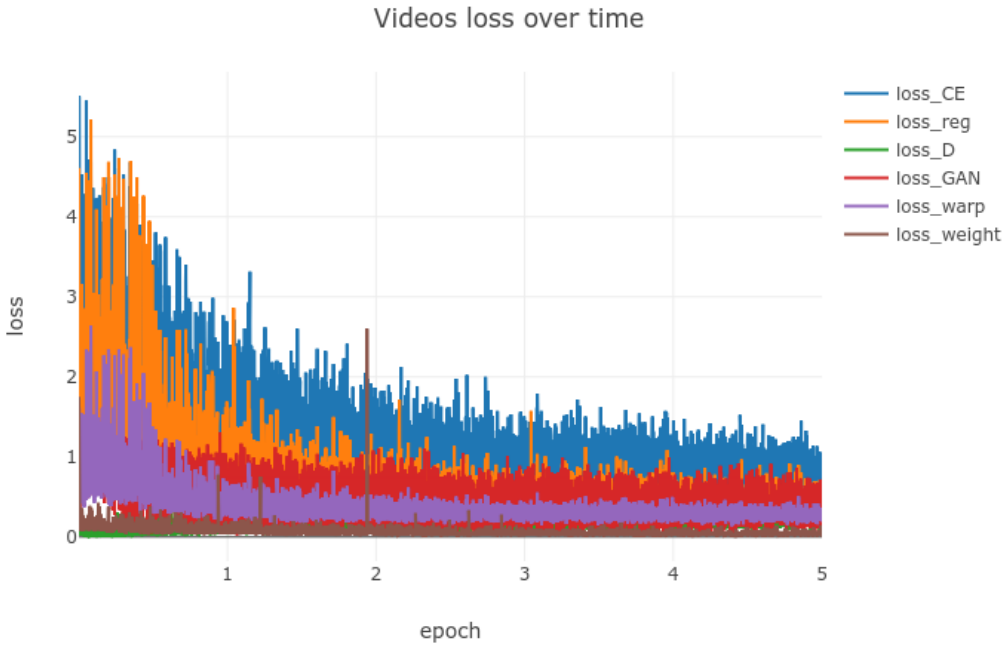


Figure 4.1. Traning Losses plot

Due to the limitation of GPUs, we set the batch size as 1, which went through one video at one iteration. In each video, we divided the image sequences into several image pairs. The image pairs contain the previous number of frames used to compute the optic flow and the next frame for colorization.

The following showed some successful results and some failure results:

(a) gray

(b) real

(c) Zhang et al.'s work

(d) Ours

Figure 4.2. Results on colorization boats

(a) gray

(b) real

(c) Zhang et al.'s work

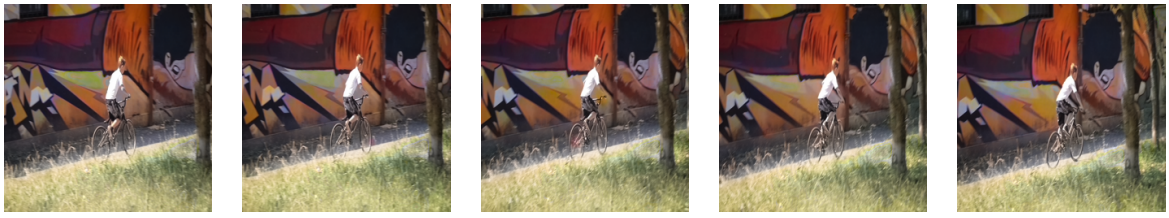
(d) Ours

Figure 4.3. Results on colorization Monkey and trees

(a) gray

(b) real

(c) Zhang et al.'s work



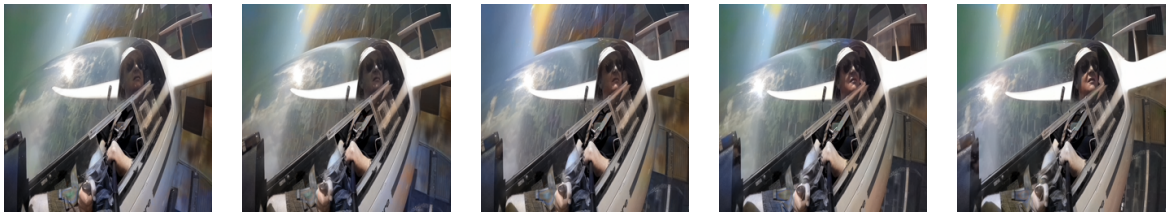
(d) Ours

Figure 4.4. Results on colorization a biker along the walls

(a) gray

(b) real

(c) Zhang et al.'s work



(d) Ours

Figure 4.5. Results on colorization a man driving a plane

(a) gray

(b) real

(c) Zhang et al.'s work

(d) Ours

Figure 4.6. Results on colorization a plane sliding on the road

4.5 Evaluation Criterion

In this thesis, we used the Peak signal-to-noise ratio(PSNR) as our criterion in evaluate our algorithm. Since we are solving the flickering problem, we used the PSNR between the adjacent frames.

$$PSNR = 20 * log_{10} \frac{Maxpixels}{\sqrt{MSE}} \tag{4.1}$$

$$MSE = \frac{1}{HW} \sum_{i=0}^{H} \sum_{j=0}^{W} (Img_1 - Img_2)^2 \tag{4.2}$$

We evaluated the results frame by frame with the ground truth. Under this circumstance, we compared grayscale images, our prediction results, and Zhang et al. 's results with the ground truth frame by frame. And we can find that our results were better than theirs.

Table 4.1. PSNR between different works

	PSNR with the ground truth
Zhang et al.'s work	27.09
Ours	**30.03**
gray	8.36

We also calculate the PSNR for each videos, which is shown as the Figure 4.7.

Among the 90 different scenes from Davis 2017, we found our prediction was good when the clips of scenes with still backgrounds. As the following features shows in Figure 4.8 and Figure 4.9,

In this clip of the scene, the background of the roads and buildings remain the same, and only the car drifted on the road. In this scene, the background buildings were colorized well, while the colored from the cars were not colorized. Compared with the prediction results from Zhang et al., our results were not flickering while they had some patches with a different color in different frames.

The following is another example of the results of colorizing a car through the dark tunnel.

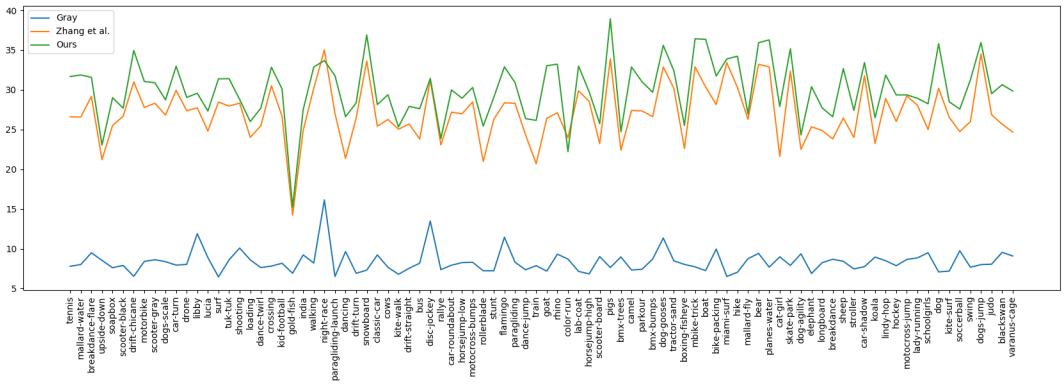


Figure 4.7. PSNR for each classes

(a) gray

(b) real

(c) Zhang et al.'s work

(d) Ours

Figure 4.8. Results on colorization of a car drifting on the road.

(a) gray

(b) real

(c) Zhang et al.'s work

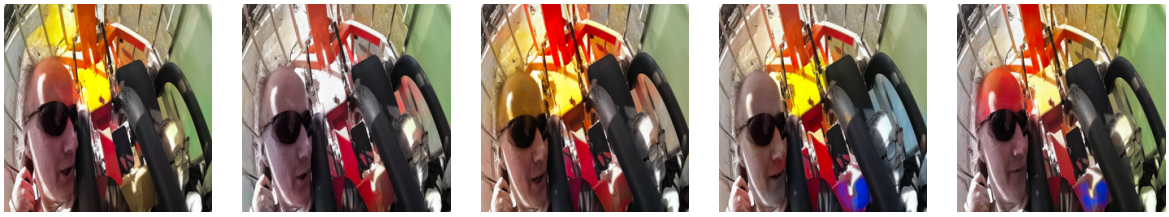
(d) Ours

Figure 4.9. Results on colorization of a car driving through the dark tunnel.

When the clip of the scene moved fast through the frames, the prediction of the colorization results still had flickering problems in our flow-guided results.

(a) gray

(b) real



(c) Zhang et al.'s work

(d) Ours

Figure 4.10. Results on colorization of a man upside-down

When the object in the clip moved too fast, our results failed to colorize it and result in some gray patches in the frames. Besides, some part was not colorized in the frames. In this clip, the yellow part from the frames was not colorized in our results. But in some frames in the compared results in Zhang et al. 's work, it colorized the color. This made the higher PSNR in their results when we evaluated the algorithm frame by frame.

We also examine our results when the objects were occluded or disappeared from the scenes. In our network design, we had a mask net to learn a hyperparameter to weight between the warped results and the generated results. In the following, we will show our result of colorizing a clip with occlude object.

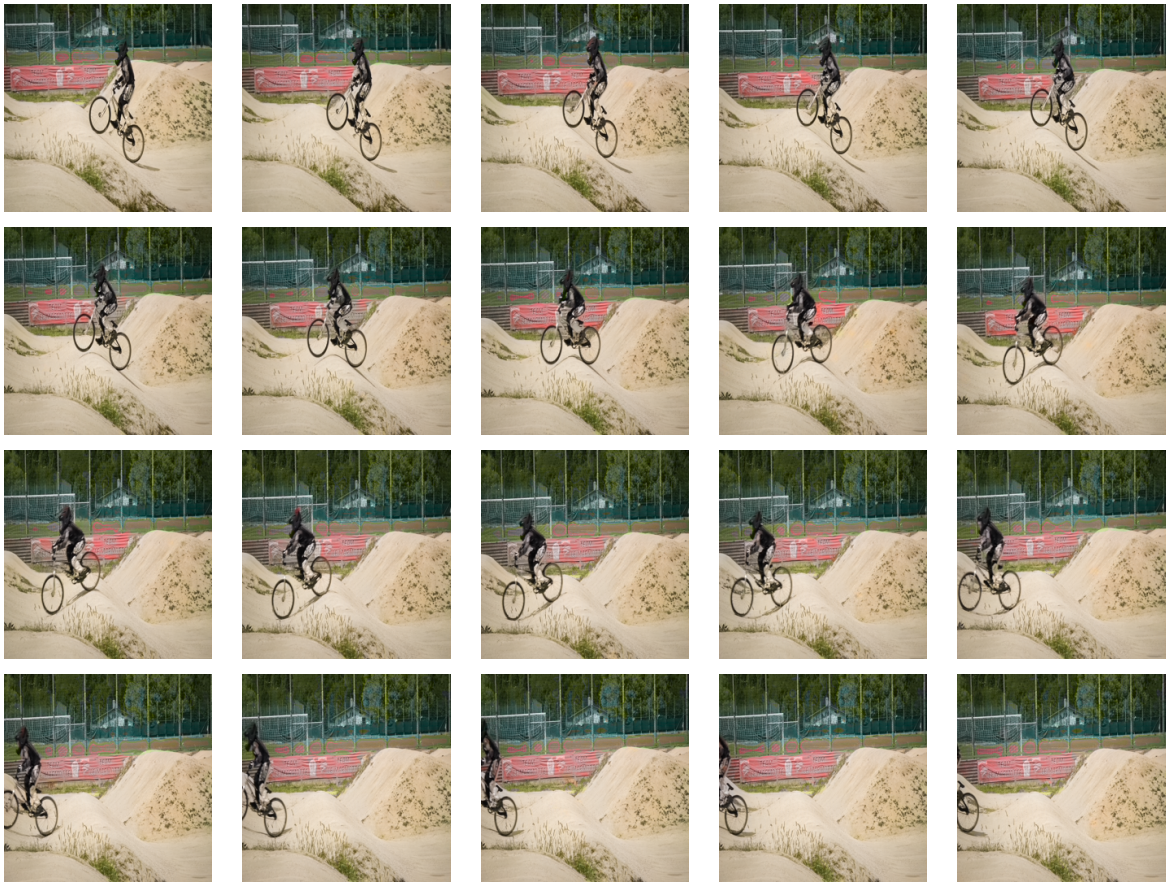


Figure 4.11. Results on colorization an occluded objects.

In this scene, the banner between the clips was occlude when a man rode across a drone. However, from the first frame and last frame in this clip, the banner was colorized almost the same. While the man rode and occluded part of the banner, the other part of the banners remained the same. Compared with the result from the frame by frame methods, the result didn't deal with the occluded objects.

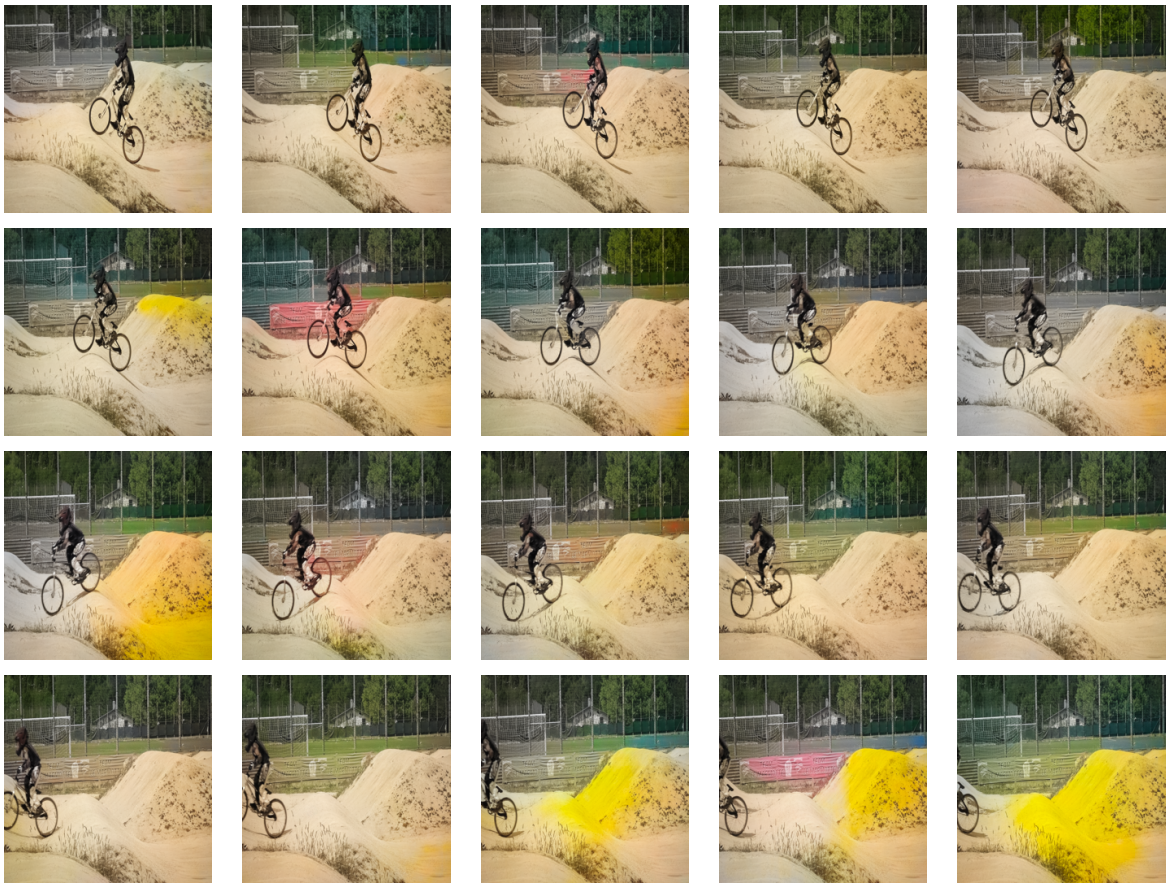


Figure 4.12. Results on colorization an occluded objects with frame-by-frame methods.

4.6 Summary

This chapter provided the details in experiment and results of this thesis.

CHAPTER 5. CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In this thesis, we presented a two-stream CNN network that employs a deep learning architecture and optic flow structures. Our approaches provide a general solution to solve the flickering problem in video-based colorization problems partially. Unlike other flow-guided problems, we also made use of the GAN structure to provide sharp and plausible results that had similar colors through the sequence of the frames. Some of our results were not as sufficient as the results from those with the frame-by-frame methods, but it gave reasonable results and kept the temporal consistency. Furthermore, we also provided a new evaluation to measure the correlation between adjacent frames and extended the PSNR to judge the performance in solving the flickering problems. Also, we used both classification and regression methods in colorizing each frame, provided more choice in the color pairs. It provided a trade-off in providing sufficient colors in the frames as the nature of multi-classes in the colors of objects in the real world and the well-known truth that a single object should have the same color in the video clips.

Our methods still suffered from fast moving objects and small objects. When the object moves fast, our flownet structure failed to capture the movement properly. That resulted in incorrect wrapping colorization results, and lead to the failure in our final prediction outputs. Besides, when the objects were small, our methods might be lost the spatial information in the down-sampling stage, where the whole object was compressed into one or two pixels. This led to the bad results in colorizing the small object in the frames, and it was one of the main reasons in the insufficient color pairs in our methods.

5.2 Future Works

Due to the limitation and still-existing problems in our methods, we can improve our approaches with the following ideas:

- we could provide a reference frame instead of the random hint color points in the generator
- we could use 3D Convolution layers instead of flow-guided structures to learn the temporal information.
- we could apply the attention transformers for both long-term and short-term memory in the sequence of frames.

REFERENCES

Chen, D., Liao, J., Yuan, L., Yu, N., & Hua, G. (2017). Coherent online video style transfer..

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., ... Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. In *Proceedings of the ieee international conference on computer vision* (pp. 2758–2766).

Gibson, J. J. (1950). The perception of the visual world.

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Humphreys, G. W., & Bruce, V. (1989). *Visual cognition: Computational, experimental and neuropsychological perspectives*. Psychology Press.

Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). 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), 110.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Ieee conference on computer vision and pattern recognition (cvpr)* (Vol. 2, p. 6).

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *arXiv preprint*.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., ... others (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Khoreva, A., Rohrbach, A., & Schiele, B. (2018). Video object segmentation with language referring expressions. In *Asian conference on computer vision (accv)*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Larsson, G., Maire, M., & Shakhnarovich, G. (2016). Learning representations for automatic colorization. In *European conference on computer vision* (pp. 577–593).

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., ... others (1995). Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks* (Vol. 60, pp. 53–60).

Luan, F., Paris, S., Shechtman, E., & Bala, K. (2017). Deep photo style transfer. *CoRR, abs/1703.07511*, *2*.

Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision.

Pascale, D. (2003). A review of rgb color spaces... from xyy to rgb. *Babel Color*, *18*, 136–152.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Robertson, A. R. (1977). The cie 1976 color-difference formulae. *Color Research & Application*, *2*(1), 7–11.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Stutz, D., Hermans, A., & Leibe, B. (2018). Superpixels: an evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, *166*, 1–27.

Wald, A. (1950). Statistical decision functions.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., & Catanzaro, B. (2018). Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*.

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision* (pp. 649–666).

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*.

Zhu, X., Wang, Y., Dai, J., Yuan, L., & Wei, Y. (2017). Flow-guided feature aggregation for video object detection. In *Proceedings of the ieee international conference on computer vision* (Vol. 3).

Zhu, X., Xiong, Y., Dai, J., Yuan, L., & Wei, Y. (2017). Deep feature flow for video recognition. In *Cvpr* (Vol. 1, p. 3).