ObstructionNet

Arpit Kumar, Syed Sameen Ahmad Rizvi, Prateek Narang

Birla Institute of Technology and Science

June 2, 2021

1 Abstract

We present a deep-learning based approach to remove occlusions such as reflections and fencing from a sequence of continuous frames in motion relative to each other. Our method uses motion difference between two images as well as extracting important moving features from the videos to separate the background and the obstruction. We introduce a novel 3D-convolution architecture and train it to separate blended videos. Further, we propose a dataset for obstruction removal that contains sequences for both reflection removal and fencing removal, which we experiment our proposed method on. The proposed method achieves state-of-the-art performance on both type of videos.

Table 1: ObstructionNet

2 Introduction

The need for removing obstruction from images/videos is of importance such as in case of reflection over glasses, obstruction due to fences/small objects. One doesn't have access to good quality photography equipment or better view, so removing obstructions from an image/set of images becomes a task to improve the overall quality of the images.
Recent efforts have been on removing reflections/obstructions from single-images[2, 8, 16, 17, 27, 38, 43, 45]. These methods utilize either ghosting cues [30] or adopt deep-learning methods to achieve the desired quality [8, 16, 38, 43, 45]. Good results can be seen in papers mentioned above but the problem here itself is problematic as they try to remove reflection/obstruction from single images with no clue about the value of the hidden pixels which can be obtained in multi-frame problems due to different angle of each frame.
To tackle these challenges, multi-frame approaches have been proposed for reflection/obstruction removal. The idea used here is the fact that the background and foreground are located at various depths with respect to the camera. They have independent movement/texture. Taking multiple slightly different frames help in getting to know the pixels occluded/less bright. A number of approaches exploit such cues for reflection or fence removal from a video [1, 3, 6, 9, 12, 21, 24, 26, 31, 34]. Xue et al[42] suggested model for the computation of background and reflection layers based on algorithmic approach of different edge-movements of elements of layers along with optimization. Recent works [1], [2] explores use of deep-learning methods for obstructions removal using multiple-frames idea. [1] uses CNN-based layered model, while [2] uses CNN-based flow-net* type architecture along with use of dense optical-field and optimization-based method for the generation of separate background and occlusion layers. while both methods produce good, comparable results over given real world sequences, they necessarily don't produce decent enough images in all cases, given the restraints of CNNs for image generation

In this work, we propose a multi-frame obstruction removal algorithm that exploits the advantages learning-based methods and dense-optical flow fields. Inspired by the learning-based approach [42], the proposed algorithm alternates between the dense motion construction and the background/obstruction layer reconstruction steps in a coarse-to-fine manner. The use of dense-motion field allows us to recognize relation between movements within background/obstruction layer. Further, we noticed the output of this layer had some minor blemishes/lack of continuity between pixels. So, we used generative networks along with above model to further improve the quality of images. We train our fusion network using a synthetically generated dataset and demonstrate it transfers well to unseen real-world sequences. Finally, we demonstrate that the proposed method performs favorably against existing algorithms on a wide variety of challenging sequences and applications.
The contributions of this work include:

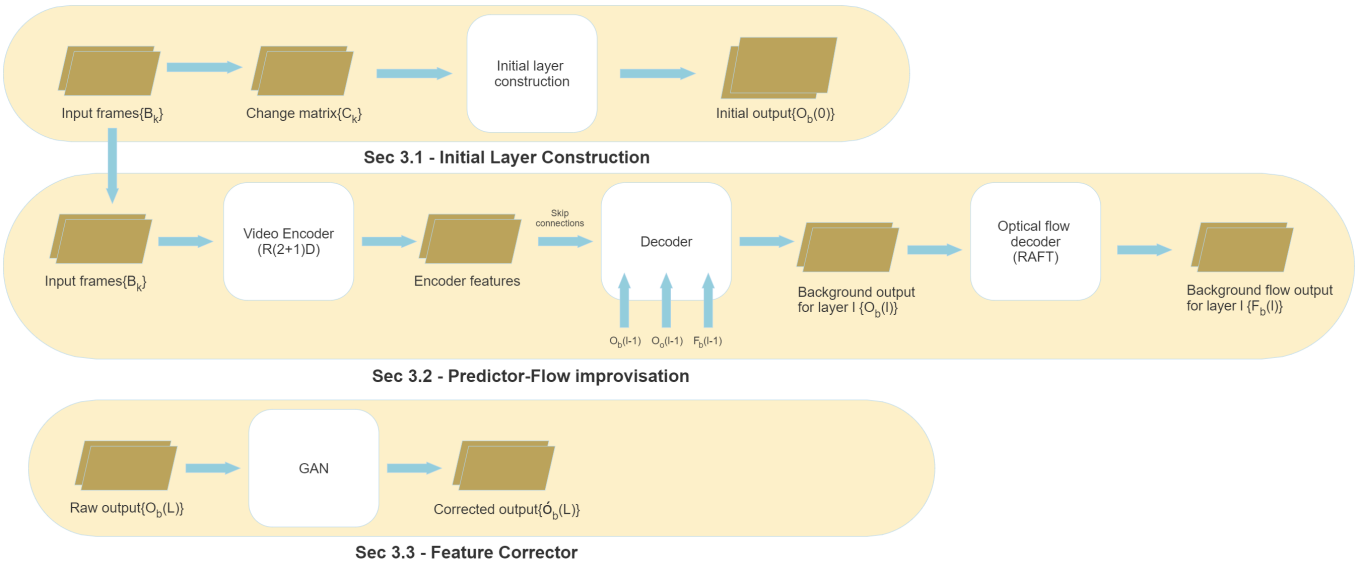
- We present a learning-based method that integrates the optical-motion based formulation for robustly reconstructing background/obstruction layers.
- We present a dataset consisting of 2000 videos of 7 frames each consisting of 1000 videos for reflection removal and 1000 videos for fencing removal.
- We show our model with minimum design changes can be applied to various obstruction removal problems.

3 Recent Works

- **Single-image reflection removal** A number of approaches have been proposed to remove unwanted reflections with only one single image as input. The existing methods exploit several different methodologies, including learned CNN methods [43, 45], use of ghosting cues that exploit asymmetry between the layer [30], regularizing the gradients of the two layers [22, 36]. Despite the amount of research carried out in the field of single-image reflection/obstruction removal, the problem is itself ill-posed as the absence of motion cues result in lack of understanding of any model of the separation of layers.
- **Multi-frame reflection removal** Existing methods often exploit the correlation of the transmitted layer across multiple images, and the sparsity and independence of the gradient fields of the two layers [12, 42] and estimating the diversities of layer motions and then recover all layers [10]. Recent advances include optimizing temporal coherence [26] and learning based layer decomposition [1]. The learning based method utilize different approaches such as dense optical-flow[*, 42], layer-based separation of videos[1]. Compared to learning layers through [*,1], our model uses both learned layer-based CNN methods combined with dense-optical flow and generative networks for better images.
- **Fence/raindrops removal** Existing methods detect fence patterns by exploiting visual parallax between fences and background scenes [25], dense flow field [42], stereo pair of fenced images in order to detect fence pixels [18]. Existing methods for raindrops removal uses attentive generative network using adversarial training[*], multi-stage architecture that progressively learns restoration functions[*] for the degraded inputs. Recent work leverages a CNN for fence segmentation [6,*] and recovers the occluded pixels using optical flow. The model presented is general in the sense it treats all kind of occlusion such as fences,raindrops,reflection the same and uses the same architecture to separate them from background.
- **Video completion** Video completion aims to remove objects or generate further sequences in a video. The state of the art methods [13,14,40] uses flow estimation and use of generative networks to generate results. The obstruction removal problem resembles a video completion task.
- **Layer decomposition** Image layer decomposition is a long-standing method used in computer vision, e.g., intrinsic image [4, 46], depth, normal estimation [15], relighting [7]. Our method relies heavily on the layer-by-layer decomposition method and using a pyramid like structure for coarse-to-fine development of background and obstruction layer.

4 Proposed Method

We decompose the set of continuous frames(B_k - input video with k frames) into background and obstruction layer using three separate layers. First is initial layer-reconstruction that reconstructs a crude background and obstruction layer from the video. Second is Predictor-Flow Improvisation that takes the output of the first layer and uses an encoder-decoder style 3D Conv-Net network along with flow-decomposition model to predict final background and obstruction



layers. In this layer, we reconstruct the output images in a coarse-to-fine manner within an L-level hierarchy. Third is Feature corrector layer that removes blemishes/pixelations and further improves the background/obstruction quality.

4.1 Initial Layer Construction

We predict both the background and obstruction layers at the coarsest layer using a 3D convolution network consisting of a Flow-Net* kind of architecture detailed in Appendix. The layer consist of two parts,

1. A encoder 3D-conv net feature extractor, to which input is concatenated numpy array of B[i], B[i+1] and change matrix. The change matrix(CH) is cross-correlation between two 2-dimensional matrix. The CH matrix helps the model gain a perspective on how the distinguish the two frames based on motion of different elements between them. The correlation between consecutive frames helps in correlation of objects in frames such that it is easy to separate two layers as the elements found correlated between two frames must belong to 1 single layer i.e. either background or obstruction.

   $$Input(i) = B[i].B[i+1].CH[i],\ Input\ is\ a\ 6-layer\ matrix.$$

   The 2-D cross-correlation of an M-by-N matrix, X, and a P-by-Q matrix, H, is a matrix, C, of size M+P-1 by N+Q-1. It's elements are given by,

   $$C(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(m,n)\tilde{H}(m-k, n-l)$$

   where

   $$-(P-1) \leq k \leq M-1$$
   $$-(Q-1) \leq l \leq N-1$$

   where the bar over H denotes complex conjugation.

2. A decoder 3D-convolution layer estimator that converts the output of feature extractor layer to the final output layer of either background or foreground detailed in Appendix.

The background and foreground differ in both the reflection and the fencing task because in reflection, the obstruction is dim as compared to the background and in fencing, the obstruction is a fence. Due to the reasons mentioned above, it was decided to use separate layer for background and obstruction with same architecture but no weight sharing.

4.2 Predictor-Flow Improvisation

We reconstruct background/obstruction in coarse-to-fine manner with the output being improved each layer after layer. Each layer(l) consists of three-sub layers

1. the video encoder layer whose aim is to extract features from continuous frames. The encoder used is a R(2+1)D* architecture, because of its use of 3D convolutions filters to separate spatial and temporal components of the video. This encoder model has shown success in action recognition from videos, which in turn helps in recognizing elemental motion frame-by-frame.

2. The outputs from (1) is used as input to the decoder model, along with the background and obstruction layer output from previous layer(l-1) and the output from (3) as skip connections into the decoder model. For the decoder, we propose a simple architecture which consists of a succession of 3D Transposed Convolutions, that has proved its worth in many dense reconstruction tasks, that we detail in Appendix.

3. Flow-estimate layer that estimates the dense-optical flow from the model from the background/obstruction layer. This layer uses RAFT-net* for the purpose, as RAFT extracts per-pixel features, builds multi-width 4D correlation volumes for all pairs of pixels, and iteratively updates a flow field through a recurrent unit that performs look ups on the correlation volumes. RAFT gives state-of-the-art performance for optical flow, and that is why we have used the model for determining the optical flow of background/obstruction. The purpose of using optical-flow as input is that it assist in determining the separate and independent movement of the two intermixed layers.

For the results, we take the number of layer as 8(L). Similar to Initial-layer Construction, we use a separate layer with same architecture for background and obstruction layer reconstruction. They do not share weights, but each use the output from the previous layer of other component as input. (i.e. background predict-flow improvisation use output of previous layer obstruction predict-flow improvisation as input). For further clarification, please see the Fig.*

$$O_b(l) = G(B_k, O_b(l-1), O_o(l-1), F_b(l-1))$$

where O_b(l) is the output of layer l for background generation predictor-flow improvisation, F_b(l-1) is the output of flow-estimate sub-layer for background generation predictor-flow improvisation for layer l-1. Here, l-1 for l=0 is the output of initial-layer construction. G represents the predictor-flow improvisation sub-layer function.

4.3 Feature Corrector

The final part of the model consists of a feature corrector. The aim of the this layer is to give final touches and correct minor blemishes/irregularities in the frames. Also, the layer helps in filling in the pixels which remain occluded throughout the video and cannot be recovered by the two previous layers. The layer uses a pix2pix* GAN architecture. Pix2pix is a cGAN that is used for image-to-image translation where a target image is generated, conditional on a given input image. The pix2pix architecture consists of 2 parts -

1. The U-Net generator model architecture that is very similar to encoder-decoder model in that it involves down-sampling to a bottleneck and upsampling again to an output image, but links or skip-connections are made between layers of the same size in the encoder and the decoder, allowing the bottleneck to be circumvented.

2. A PatchGAN discriminator – that only penalizes structure at the scale of patches. The discriminator tries to classify if each NxN patch in an image is real or fake. We run this discriminator convolutionally across the image, averaging all responses to provide the ultimate output of D.

Due to the constraints of GAN that only accepts 1 image at a time, all the frames are concatenated together and given input as 1 image({W, H*f}, where W is width, H is height and f is no of frames). The output obtained is then converted back into separate frames(W,H).

4.4 Network Training

To train the complete model, we train the model layer-by-layer. First, the Initial-Layer Construction is trained using the following loss

$$L = |O_b(0) - V_b|^2$$

where V_b is the actual background output frame. The above example is for background layer construction architecture. Similar loss is used for obstruction architecture also.

Next, we freeze the Initial-Layer Construction layer and use it's output as input to Predictor-Flow Improvisation layer, which is trained using the following loss for each layer(l)

$$L = \sum_{l=0}^{l=L} |O_b(l) - V_b|^2 + |F_{\bar{b}}(l) - F_b(l)|^2$$

where $F_{\bar{b}}(l)$ is the actual background motion flow and $F_b(l)$ is the predicted background motion flow for layer l. The output of the Predictor-flow improvisation layer is used as input for feature corrector layer. The loss used is a traditional cGAN loss with L1 used as a sidekick to the adversarial loss function.

$$\mathcal{L}_{\text{cGAN}}(G, D) = \mathcal{E}_{x,y}[logD(x, y)] + \mathcal{E}_{x,z}[log(1 - D(x, G(x, z)))]$$

The L1 loss mentioned is

$$\mathcal{L}_{L1}(G) = \mathcal{E}(x, y, z)[||y - G(x, z)||]$$

Combining these functions results in :

$$G* = \underset{G}{argmin}\underset{D}{max}\mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G)$$

The value of λ parameter is taken to be 100 as reported by the authors of the paper*. The discriminator model is trained directly on real and generated images, whereas the generator model is not. Instead, the generator model is trained via the discriminator model. It is updated to minimize the loss predicted by the discriminator for generated images marked as "real." As such, it is encouraged to generate more real images. The generator is also updated to minimize the L1 loss or mean absolute error between the generated image and the target image.

4.5 Synthetic Dataset Generation

Since collecting real sequences with ground-truth obstruction and background layers is very difficult, we use Vimeo-90k* along with De-fencing* dataset to synthesize sequences for training.

1. For reflection, we combine two, 7 frame video randomly selected from Vimeo-90k 8k test video. The two videos are combined linearly using α value selected randomly between 0.5 - 0.7 with background being greater or equally brighter than the foreground/obstruction. Formally, given two videos, $V^1, V^2 \in R^{T \times H \times W \times 3}$, where T is the total number of frames, H and W the frame's height and 3 corresponds to the standard RGB channels, we generate a training video V as follows:
$$V = \alpha V^1 + (1 - \alpha)V^2$$
2. For fencing, we again combine one 7 frame video randomly selected from Vimeo-90k 8k test video with 7 continuous fence frames from De-fencing dataset. Uniformly, the fencing videos are generated with static fence with independent moving background and moving fence with moving background, both in motion independently.

The size of frames is kept uniformly at 896x512 pixels for all reflection and fencing videos. The entire dataset can be found on the project page.

5 Experiment

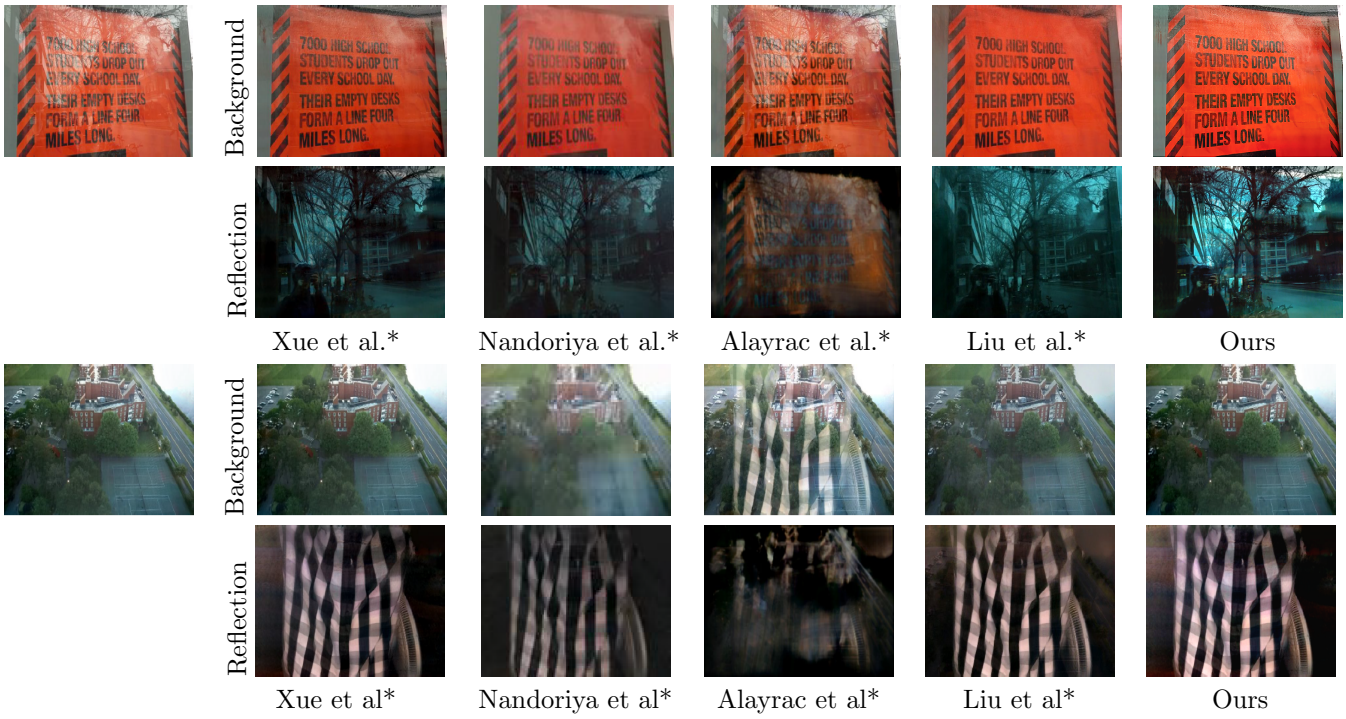
The models were trained on GeForce RTX 2080 GPU. Initial-layer Construction network was trained for 20k iterations separately for reflection and fencing task. Each task had 2 sub-task i.e. separate background and reflection model for each reflection/fencing task. The Predictor-Flow improvisation network was run for 50k iterations. We use Stochastic Gradient Descent(lr = 0.01 for first 20k and 0.001 for next 30k) along with norm-clipping for training. The encoder model(R(2+1)D*) and Optical-flow decomposition model(RAFT-net*) are used as pre-trained models. The Feature corrector model was trained for 20k iterations.

6 Analysis and Discussion

For analysis of our performance, we compare our model on two types of sequences -

Method			Reflection				Background			
			PSNR↑	SSIM↑	NCC↑	LMSE↓	PSNR↑	SSIM↑	NCC↑	LMSE↓
Single image	CEILNet	CNN-based	20.35	0.7429	0.8547	0.0277	-	-	-	-
	Zhang et al.	CNN-based	19.53	0.7584	0.8526	0.0207	18.69	0.4945	0.6283	0.1108
	BDN	CNN-based	17.08	0.7163	0.7669	0.0288	-	-	-	-
	ERRNet	CNN-based	22.42	0.8192	0.8759	0.0177	-	-	-	-
	Jin et al.	CNN-based	18.65	0.7597	0.7872	0.0218	11.44	0.3607	0.4606	0.115
Multiple images	Li and Brown	Optimization-based	17.12	0.6367	0.6673	0.0604	7.68	0.267	0.349	0.1214
	Guo et al.	Optimization-based	14.58	0.5077	0.5802	0.0694	14.12	0.315	0.3516	0.1774
	Alayrac et al.	CNN-based	23.62	0.7867	0.9023	0.02	21.18	0.632	0.7535	0.1517
	Liu et al.	CNN-based	**26.57**	0.8676	**0.938**	**0.0125**	21.42	0.6438	**0.7613**	**0.1008**
	Ours	CNN/GAN-based	24.08	**0.890**	0.881	0.022	**21.81**	**0.66**	0.75	0.138

Table 2: The numbers are taken directly from *. The authors synthesized 100 sequences from the Vimeo-90k test set. We also use the same 100 sequences and compare our approach with five single-image reflection removal methods [8, 16, 38, 43, 45], and four multi-frame approaches [1, 12, 21, *].



1. Synthetic sequences - We obtained 100 sequence from the authors of *. The sequences were synthesized by the method described in the paper from the Vimeo-90k test set. We compare our approach with five single-image reflection removal methods [8, 16, 38, 43, 45], and four multi-frame approaches [1, 12, 21, *]. Since, the sequences are same as used in *, we directly take the data for above mentioned papers from their publication. Table * shows the average PSNR, SSIM [37], NCC, and LMSE [11] metrics. THE PROPOSED METHOD OBTAINS THE BEST SCORES ON ALL THE EVALUATION METRICS FOR BOTH BACKGROUND AND REFLECTION LAYERS and clearly outperforms the existing models in the problem.

2. Real sequences. In Figure 5, we present visual comparisons of real input sequences from [42]. Our method is able to separate the reflection layers and reconstruct comparable clear and sharp background images to other approaches [1, 21, 26, 42, *]. Our method can separate the obstruction and background layer and reconstruct clear background images. Further generated image sequences have been shown in the appendix of the paper.

We also analyse the model layer wise and analyse the effect of each layer on the results. Since, output of each layer is a 6-frame video, each layer can be separately analysed on the effect it has on the overall results. We use PSNR value for evaluation of each layer with the ground truth values for a test video. The sample generated images for background frames are also displayed with the results.

- **Initial layer construction** - The aim of the layer was create an initial estimation of the background/obstruction layer and separating some major elements of each layer. The layer used encoder-decoder architecture with correlation data(Change MatrixCH) between two consecutive frames to support it. The correlation between

consecutive frames helps in correlation of objects in frames such that it is easy to separate two layers as the elements found correlated between two frames must belong to 1 single layer i.e. either background or obstruction.

- **Predictor-flow improvisation** - The layer uses both CNN-based and dense-flow based methods for separation of layers. This layer is the backbone of the method and majority of the work is expected of it. The pyramid type architecture used such that it helps in generation of layers in a coarse-to-fine manner. First, the major elements of the layer are separated followed by the smaller elements. The layer is the largest in terms of weights and number of epochs.
- **Feature corrector** - This layer is put forward for minor correction in frames such as any blemishes or lack of discontinuity in any layer produced by the above two layer. Also, the layer helps in filling in the pixels which remain occluded throughout the video and cannot be recovered by the two previous layers. This layer, as expected, produced only a minor improvement in the overall result.

Layer	**PSNR**
Initial layer construction	13.34
Predictor-flow improvisation	22.128
Feature corrector	23.59

7 Conclusion

We have presented a novel method for multi-frame reflections and obstructions removal. Our key insight is to leverage a CNN to reconstruct background and reflection layers from images. Integrating optical flow usage and coarse-to-fine refinement enable our model to robustly recover the underlying clean image from challenging real-world sequences. Our method can be applied to different tasks such as fence or raindrop removal with minimum changes in our design. Extensive visual comparisons and quantitative evaluation demonstrate that our approach performs well on a wide variety of scenes.

8 Appendix

8.1 Model Architectures

Figure 1: Initial Flow construction

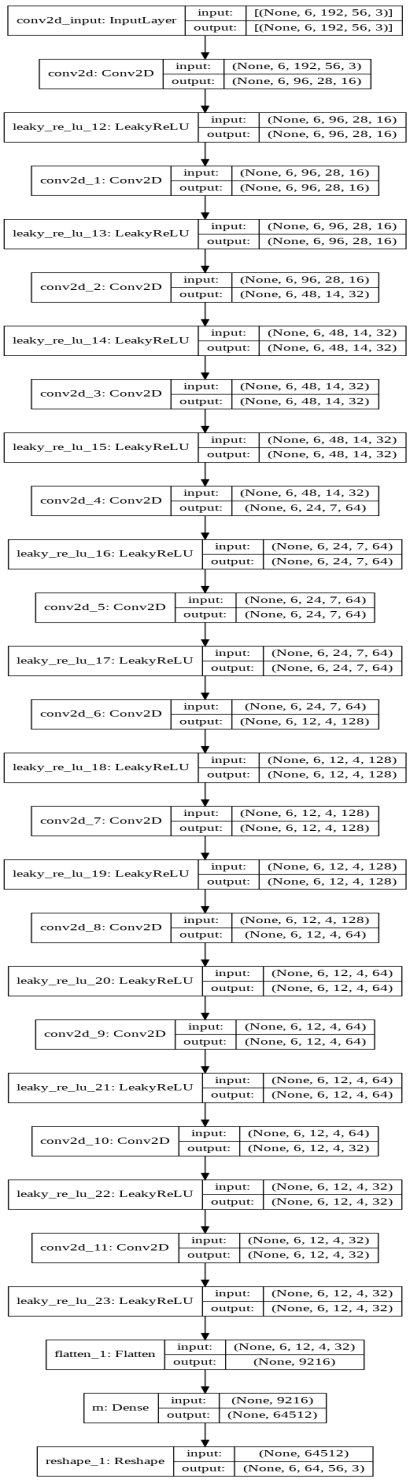
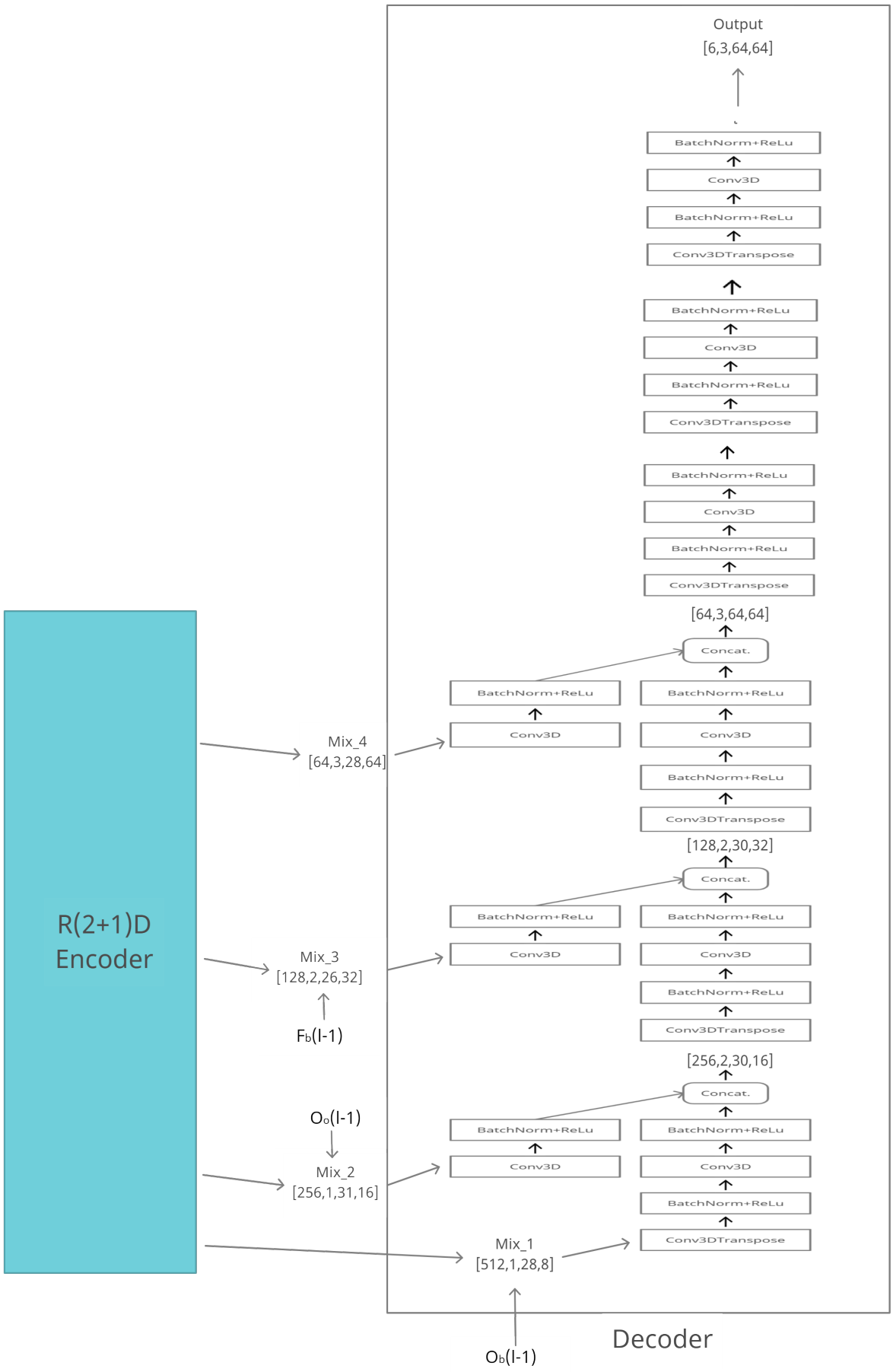
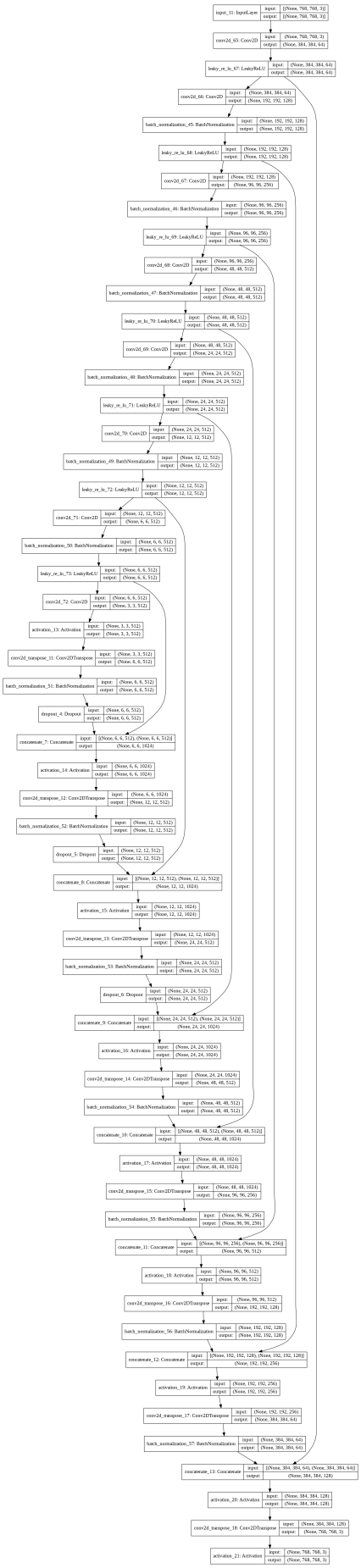


Figure 2: Predictor-Flow improvisation layer



(a) Feature corrector layer - generator



(b) Feature corrector layer - discriminator

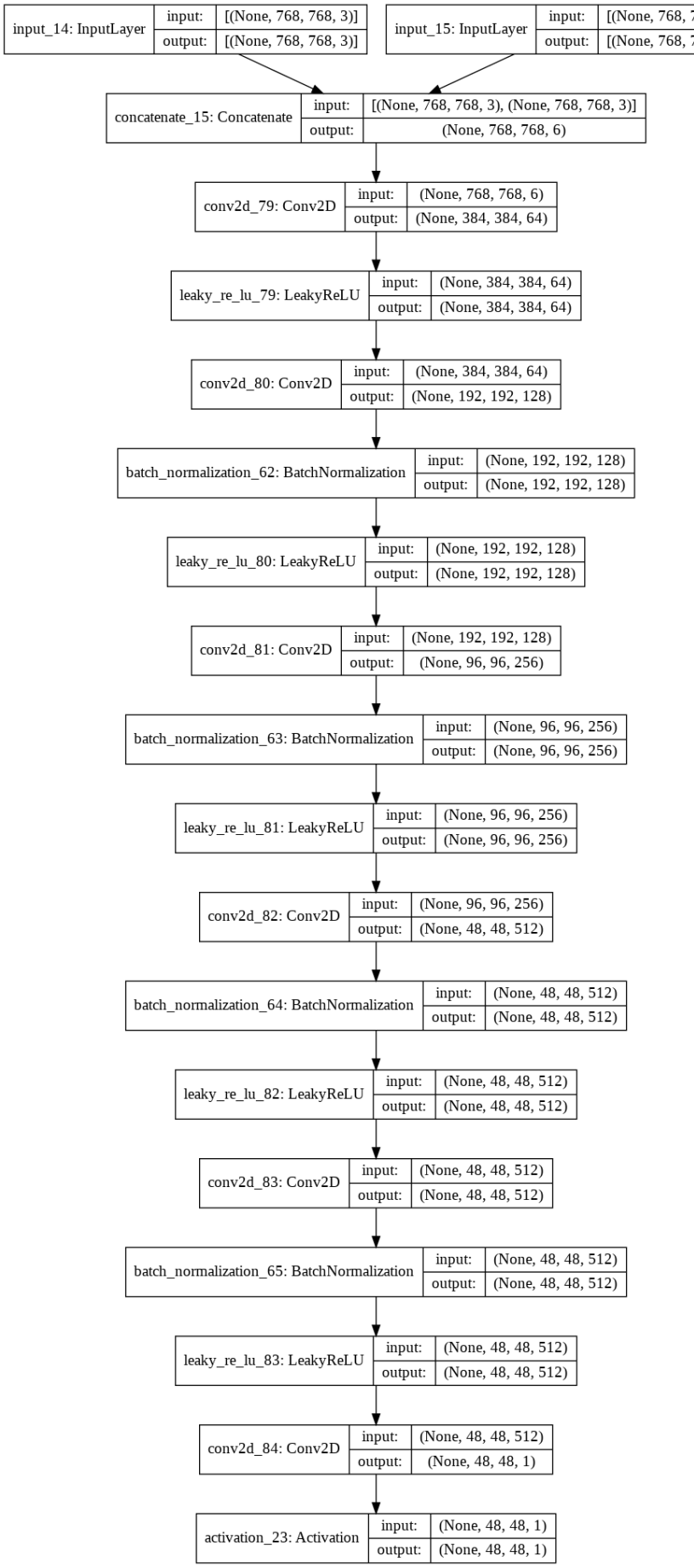


Figure 3: Architecture for General Adversarial Networks