DRPI-CGAN: Detection and Restoration of Photoshopped Images using Convolutional GAN

Mingxuan Teng
University of Toronto
mingxuan.teng@mail.utoronto.ca

Jiaming Yang
University of Toronto
jiaming.yang@mail.utoronto.ca

Jiakai Shi
University of Toronto
jiakai.shi@mail.utoronto.ca

1 Abstract

Nowadays, images can “lie”. This phenomenon is especially common in social media websites, where faces are always fabricated. Usually, those manipulations are hard to detect by human eyes. In this report, we present a method based on Convolutional Generative Adversarial Network (Conv GAN) to detect and undo those manipulations on human faces. We show that our model performs well for both detecting and "undoing" the manipulations. We demonstrate why our method provides a possible way for solving problems of facial image manipulation detection and recovery. We also discuss about the limitations of our model and the possible future steps for improvements.

2 Introduction

Recent work by a research team of UC Berkeley and Adobe Research called FALDetector has presented an excellent model in detecting and "undoing" manipulations on images containing faces [1]. In their work, they transform images to optical flows so that they can tackle the problem in the "flow space". They use a fine-tuned Dilated Residual Network (DRN-C-26) to classify whether an image is warpped and to localize the warpped parts of an image. After generating a reliable predict flow, they apply are reverse transformation to undo those "warps". Although their method already has a promising result and outperforms human at recognizing manipulated images, we believe the performance of their model can still be improved by taking part of their model and combining it with a Conv GAN structure. We think that rather than undoing the warps by editing the warped images, we can reverse the warp manipulations by generating a fake image that is very close to the unwarpped target. Similar to FALDetector, we also want to detect and "undo" the warps in flow space. But instead of the traditional way of training one network, we want to train the model through a adversarial process by two networks. Specifically, we take the DRN-C-26 as the generator of our Conv GAN. During the training process, it consistently generates "fake" flow to see whether it is able to trick the discriminator to believe that this fake flow is generated from the unwarpped target. Through out the adversarial process, both generator (DRN-C-26) and disriminator (Deep Conv-Net) improves a lot so that we also gets promising results of detecting and "undoing" the warps. The details of our model are discussed in the **Method** section and the results are presented in **Experimental Results** section.

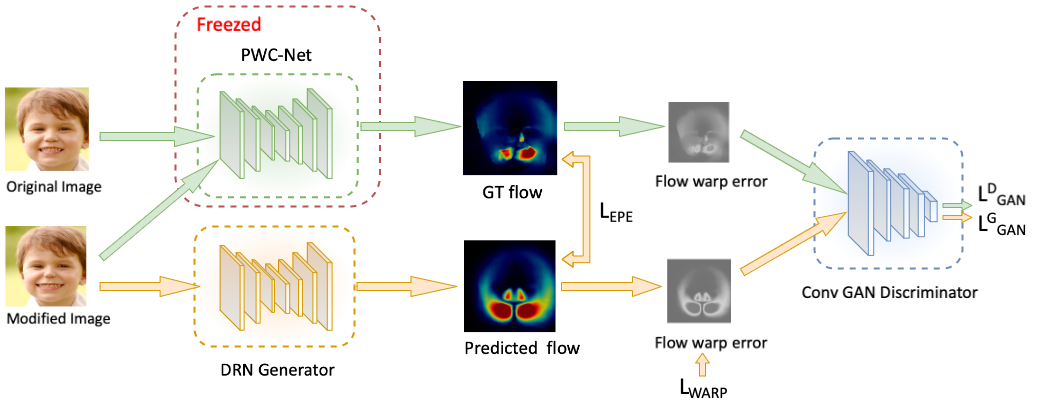


Figure 1: DRPI-CGAN semantics diagram. Parameters in PWC-Net are freezed since we only used to pretrained model to generate optical flow from modified image to original image as our Ground Truth (GT) flow. PWC-Net is chosen due to its excellent performance in estimating optical flow in sequencing images [9].

3 Method

In this section, we introduce our purposed method. We first introduce the constructed objective function including end-to-end point error loss, warpping loss and adversarial loss. Then we provide an introduction of the architecture of DRPI-CGAN and describe each component in detail. An overview of DRPI-CGAN structure is shown in Figure 1.

3.1 Objective function

End-to-end Point Error Loss. To address the problem of optical flow differences, we construct the end-to-end point error loss as follows [11]: Given a modified image $I_m \in \mathbb{R}^{3\times H\times W}$ and a original image $I_o \in \mathbb{R}^{3\times H\times W}$, we predict the optical flow $f_{pred} = [u_{pred}, v_{pred}] \in \mathbb{R}^{2\times H\times W}$ through DRN Generator and use PWC-Net to produce a ground truth optical flow $f_{gt} = [u_{gt}, v_{gt}] \in \mathbb{R}^{2\times H\times W}$ [9]. We minimize the EPE loss between two flows:

$$L_{\text{EPE}}(f_{pred}, f_{gt}) = \sqrt{(u_{gt} - u_{pred})^2 + (v_{gt} - v_{pred})^2} \tag{1}$$

Warpping Loss. We further construct a warpping loss to assist the optimization of the generator:

$$L_{\text{WARP}}(f_{pred}, I_m, I_o) = I_o - \mathcal{W}(f_{pred}, I_m) \tag{2}$$

where $\mathcal{W}(f, I)$ is a function which warps the image I with an optical flow f [11]. The warpping loss is also called flow warp errors in Figure 1, where the GT flow warp error is $e_{gt} = I_o - \mathcal{W}(f_{gt}, I_m) \in \mathbb{R}^{3\times H\times W}$ and the predicted flow warp error is $e_{pred} = I_o - \mathcal{W}(f_{pred}, I_m) \in \mathbb{R}^{3\times H\times W}$. The flow warp error pair (e_{pred}, e_{gt}) is used in the discriminator during the adversarial training.

Adversarial Loss. The main idea behind Generative Adversarial Networks (GAN) comes from the basic principle of a two-player zero-sum game, i.e. there are two networks, a generator and a discriminator, balance their loss to each other in a minmax process and finally reach Nash equilibrium [2, 3]. To update generator G and discriminator D respectively, we construct our losses:

$$L^G_{\text{GAN}}(e_{pred}) = L_{\text{BCE}}(D(e_{pred}), 1) = -\log(D(e_{pred})) \tag{3}$$

$$\begin{aligned} L^D_{\text{GAN}}(e_{pred}, e_{gt}) &= \frac{1}{2}\Big(L_{\text{BCE}}(D(e_{gt}), 1) + L_{\text{BCE}}(D(e_{pred}), 0)\Big) \\ &= -\frac{1}{2}\Big(\log(D(e_{gt})) + \log(1 - D(e_{pred}))\Big) \end{aligned} \tag{4}$$

We incorporate end-to-end point error loss and warpping loss to adversarial loss and solve the minmax problem of optimizing G and D as our objective function:

$$Obj(D, G) = \min_G \max_D \lambda_1 L_{\text{EPE}}(G) + \lambda_2 L_{\text{WARP}}(G) + \lambda_3 L^G_{\text{GAN}}(G) + \lambda_4 L^D_{\text{GAN}}(D) \tag{5}$$

where we set $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ to $\{0.1, 0.01, 0.1, 10\}$ respectively for our experiment.

3.2 DRPI-CGAN architecture

Generator. Our goal is to generate optical flow to approach the ground truth optical flow from PWC-Net [1, 9]. We choose Dilated Residual Network variant (DRN-C-26) as our generator since it outperforms ResNet-101 and its variants in semantic segmentation [10]. We find that the Dilated Residual Network architecture not only works well in semantic segmentation, but also benefits the optical flow estimation [1, 10].

Discriminator. The goal of discriminator is to identify the predicted flow coming from the generator as fake [12]. To do this, we convert both predicted flow and GT flow to flow warp errors and pass them to the discriminator as inputs. The discriminator is constructed with convolutional layers with LeakyReLU activation functions as intermediate for superior performance than regular ReLUs [7, 8]. Like the discriminator in DCGAN, the convolutional layers are used as feature extractors [7]. However, we modified the discriminator from DCGAN where the discriminator outputs a feature map of prediction rather than outputting a binary value to identify whether the flow warp error is real or not. We find that the feature map helps with alleviating the typical "symptom" in GAN - discriminator loss L^D_{GAN} converging quickly to zero - so that it provide a reliable path for updating the generator [8, 13].

Implementation details. We implement our purposed method by using Torch framework [14]. We use Adam optimizer for updating generator and discriminator, and we set the learning rate to 10^{-4} [15]. We update generator and discriminator separately in each training iteration, and we train the network for a total of 34k iterations.

We use Flickr-Faces-HQ Dataset (FFHQ) on Kaggle as our original images [1]. This dataset is created for benchmark in StyleGAN and it contains high-quality images [17]. Then we used a photoshop script for automatically generating modified images from original. We choose photoshop as our tool for manipulating images because it is the most popular software for editing images [16]. We use 8752 images from FFHQ and create corresponding modified images for training. Then we use another 500 images as the validation set for later evaluation. Out source code is publicly available on GitHub: `https://github.com/VictorS67/DRPI-CGAN`.

4 Experimental Results

Evaluations. In general, we apply three metrics to measure the similarity between predict flow and ground truth flow. To be more specific, these three metrics are: (1) End-to-end Point Error (EPE): $\|f_{pred} - f_{gt}\|_2$ evaluates the similarity between the predict flow f_{pred} and GT flow f_{gt}. (2) Intersection over Union (IoU$_\tau$): $\frac{\|f_{pred} \cap f_{gt}\|}{\|f_{pred} \cup f_{gt}\|}$ and a threshold τ is used to determine the bounding box of the magnitude of predicted flow and GT flow. We set $\tau = 0.5$ where the magnitude greater than 0.5 is considered inside the bounding box. (3) Delta Peak Signal-to-Noise Ratio (ΔPSNR): PSNR$(I_o, \mathcal{W}(f_{pred}, I_m))$ - PSNR(I_o, I_m) evaluates the effectiveness of unwarpping.

Evaluating Results. We show our results from evaluation metrics in the Appendix. From Figure 3, it can be seen that our model is consistently making progress as the training proceeds. Figure 3a and 3b show the EPE and ΔPSNR results of our model at different stages during training process. It can be seen that EPE sharply converges from 47.8 to 2.2 at around 6k training steps and remains decreasing at a small rate afterwards. In addition, the trend for ΔPSNR corresponds to that of EPE. ΔPSNR has a sharp increasing trend before approximately 6k iterations and increase slowly after that point. Both EPE and ΔPSNR results show one direction convergent trends, which means the predict flow is converging to the ground truth flow. In Figure 3c, it can also be seen that our model is improved by training - the IoU result has a clear increasing trend after 6k iterations. This shows that our predict flow is improved by containing more elements/features that are in ground truth flow. However, IoU may not be an accurate metric during all of the training stages. We could observe that the value of IoU is high at the beginning of the training. Normally, a high IoU value (e.g. IoU$_\tau > 0.5$) indicates good model performance that the predict flow is close to the ground truth flow. However, such analysis does not match our observations in EPE and ΔPSNR at early stage of training process, where our model is not capable to generate such high quality flows. This contradiction is expected since predicted flows are mostly random and cover almost all of the image area before 6k training steps. However, IoU

Table 1: Performance comparison between DRPI-CGAN and FALDetector.

Model	EPE	Δ PSNR	$\text{IoU}_{0.5}$
DRPI-CGAN	1.7125	**-11.0661**	0.7146
FAL	**0.8926**	-11.0676	**0.8357**

would be a good evaluation metric as the EPE and ΔPSNR results are converging to the optimal.

Training Results. The overall results of training is shown in Figure 2 in Appendix. In Figure 2c and 2e, L^G_{GAN} and L^D_{GAN} show a clear convergent trend with fluctuations. This trend of our model is very similar to loss trend in DCGAN [7], which provide evidence that our GAN structure is functioning as expected. In Figure 2a and 2b, the total loss of generator follows the same pattern as L_{EPE}. This observation is expected since we set the weight of L_{EPE} to be the main contributor of total loss of generator. Both of the total generator loss and L_{EPE} converge to a relatively low value very fast until they reach a point where the loss can hardly decrease. This implies that there should be another loss existed in the process of transferring from image space to flow space. We try to construct L_{WARP} to reflect such loss. In Figure 2d, however, the L_{WARP} never converges through out the whole training process. This means that the model is not consistently improved in "image-flow" transformation. As a result, the existence of L_{WARP} prevents the L_{EPE} between the predict flow and ground truth from converging.

Model Comparison. We compare the overall performance of our model with FALDetector. We evaluate the pretrained FALDetector with 500 image pairs in the validation set and compare these metric results with the final evaluation results in our DRPI-CGAN model, as shown in Table 1. Although our model has a similar level of performance as FALDetector in ΔPSNR, FALDetector outperforms our model in EPE and $\text{IoU}_{0.5}$. We examined performance difference between DRPI-CGAN model and FALDetector in Table 2 and Table 3.

Limitations & Future Works. Although our model does not outperforms the FALDetector, we think there are some possible reasons and potential future works: (1) As shown in Figure 2d, the L_{WARP} only fluctuates without converging. We find that even the GT flow cannot unwarp modified images to the exact same as the original image. To solve this problem, we want to add a "forward-backward consistency" test for training so that we can ensure L_{WARP} loss to be able to measure the flow quality with consistency; (2) For each iteration, we input a new image to our generator to generate the predict flow. Since flows are remarkably different for different images, frequent changes of input images may break the consistency of model on learning warping features. We want to explore other loss methods to address this problem in the future; (3) The current training size is insufficient for our model. FALDetector is trained with 300k iterations to get the promising result, while our model only trains for 34k iteration. We believe that training more iterations will help to improve the performance of our model since previous inquiry shows that our model is still consistently making progress. Our model has not reached its maximum performance.

5 Conclusion

We have presented a method based on Conv GAN structure to detect and "undo" facial warping manipulations. We show that our model have a promising performance for detecting and reversing warps. We believe that our model will be further improved in several ways: (1) applying forward-backward consistency tests to warp loss; (2) address the problem of flow feature learning with more sophisticated losses; (3) train the model with a larger training set. Moreover, we also see our work a step toward forensic tools that can securely identify human faces without influenced by image manipulations.

6 Reference

[1] Wang, Sheng-Yu, et al. "Detecting photoshopped faces by scripting photoshop." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[2] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).

[3] Wang, Kunfeng, et al. "Generative adversarial networks: introduction and outlook." IEEE/CAA Journal of Automatica Sinica 4.4 (2017): 588-598.

[4] Teddy Surya Gunawan, et al. "Development of Photo Forensics Algorithm by Detecting Photoshop Manipulation Using Error Level Analysis" Vol. 7, No. 1, July 2017, pp. 131 $\sim$ 137

[5] Vt, Manu & Mehtre, B.M.. (2018). Tamper Detection of Social Media Images using Quality Artifacts and Texture Features. Forensic Science International. 295. 10.1016/j.forsciint.2018.11.025.

[6] G. Clara Shanthi, et al. A Novel Approach for Efficient Forgery Image Detection UsingHybrid Feature Extraction and Classification. International Journal of Engineering & Technology, 7 (3.27) (2018) 215-219

[7] Li, J., Jia, J., & Xu, D. (2018, July). Unsupervised representation learning of image-based plant disease with deep convolutional generative adversarial networks. In 2018 37th Chinese control conference (CCC) (pp. 9159-9163). IEEE.

[8] Creswell, Antonia, et al. "Generative adversarial networks: An overview." IEEE Signal Processing Magazine 35.1 (2018): 53-65.

[9] Sun, Deqing, et al. "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[10] Yu, F., Koltun, V., & Funkhouser, T. (2017). Dilated residual networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 472-480).

[11] Lai, W. S., Huang, J. B., & Yang, M. H. (2017). Semi-supervised learning for optical flow with generative adversarial networks. Advances in neural information processing systems, 30.

[12] Che, T., Zheng, Y., Yang, Y., Hou, S., Jia, W., Yang, J., & Gong, C. (2021). SDOF-GAN: Symmetric Dense Optical Flow Estimation With Generative Adversarial Networks. IEEE Transactions on Image Processing, 30, 6036-6049.

[13] Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862.

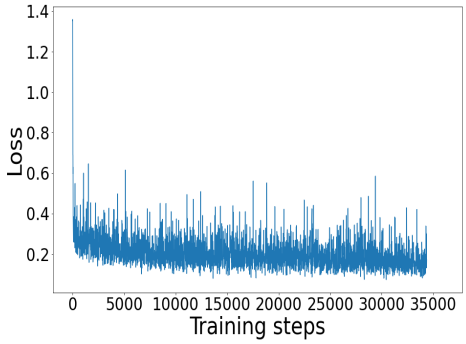
[14] Collobert, R., Kavukcuoglu, K., & Farabet, C. (2011). Torch7: A matlab-like environment for machine learning. In BigLearn, NIPS workshop (No. CONF).

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

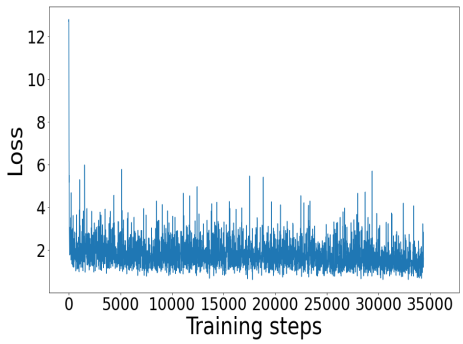
[16] Adobe Photoshop CS. (2004). Berkeley, CA: Peachpit Press.

[17] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).

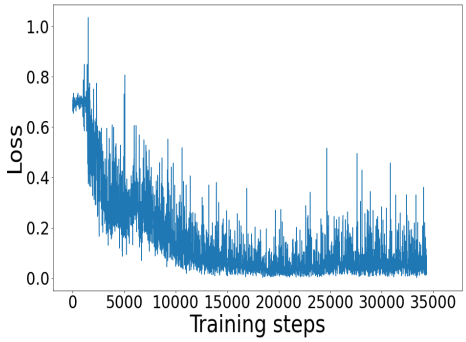
A Appendix



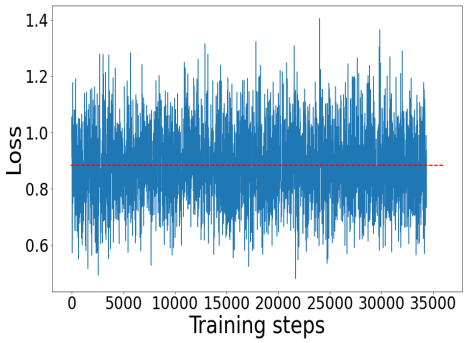
(a) Total loss for the generator.



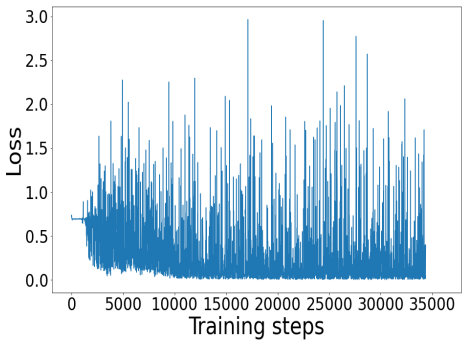
(b) EPE loss L_{EPE}.



(c) Adversarial loss L^{G}_{GAN} for generator.

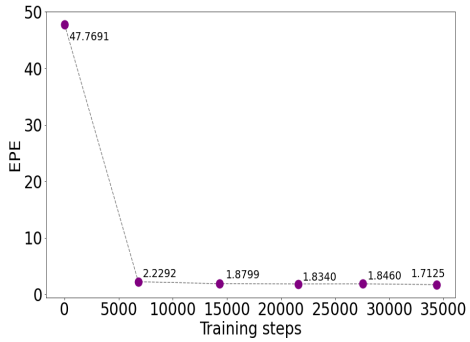


(d) Warpping loss L_{WARP}.

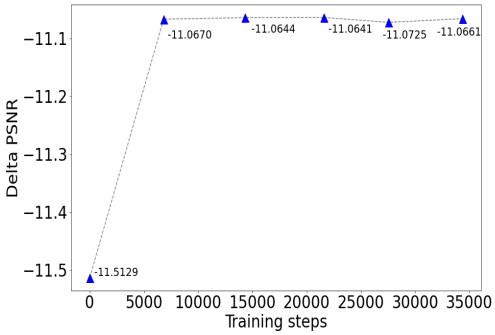


(e) Adversarial loss L^{D}_{GAN} for discriminator.

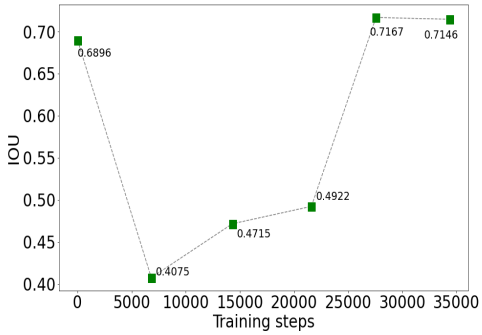
Figure 2: Different losses during training process. (1) We train the DRPI-CGAN with 8752 images in 5 groups. For each group, we train the model with $1700 \sim 1800$ different images. (2) We set the epoch to 8 and batch size to 2 for each group training. In each epoch, the model is trained with shuffled image pairs. There are 34368 training steps in total. (3) For warpping loss L_{WARP}, the red dash line indicates the mean value of the warpping losses.



(a) EPE



(b) ΔPSNR



(c) $\text{IoU}_{0.5}$

Figure 3: Three metrics used for evaluation. All three metrics imply that our DRPI-CGAN is functioning as expected. Both ΔPSNR and $\text{IoU}_{0.5}$ have forecast increasing trend and EPE has forecast decreasing trend, which means the model may perform better in later training process.

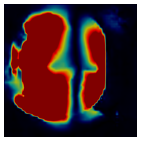
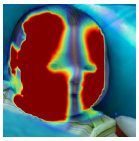
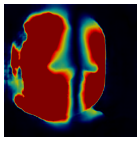
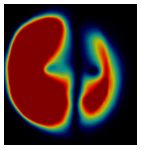
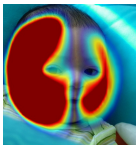
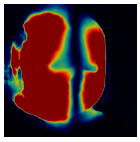
Modified image	Original image	Predicted flow	Flow overlay	Warpped image	GT flow
					
					

Table 2: We show an example of overfitting DRPI-CGAN model after 1900 iterations (first row) and compare the overfitted results with FALDetector (second row). We find that (1) DRPI-CGAN predicts almost identical optical flows as PWC-Net predicts (See first row); (2) The optical flow predicted from PWC-Net does not produce the best warpped image (See first row); (3) Compared to the PWC-Net, FALDetector predicts better optical flow where the warpped image is smoother.

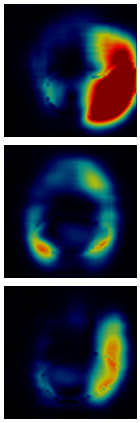
Predicted flow	Flow overlay	Warpped image	GT flow	Predicted flow	Flow overlay	Warpped image
						

Table 3: We show some examples of predicted flow, flow overlaid on modified image and warpped image from modified image by DRPI-CGAN model (left 3) and FALDetector (right 3). GT flow is the ground truth flow generated by PWC-Net (center). We find that (1) DRPI-CGAN performs well in predicting flow when the one side of the face has been manipulated, as shown in the first row; (2) DRPI-CGAN often predicts flow on the top of the face while FALDetector predicts on two sides of the face, as shown in the second row; (3) DRPI-CGAN performs badly when the manipulation is very obvious on two sides of the face but FALDetector performs well, as shown in the third row.

B Contribution

B.1 By Type

- **Designed and built DRPI-CGAN infrastructure:** Jiakai Shi, Mingxuan Teng.
- **Designed, implemented data pipeline:** Jiakai Shi.
- **Designed, implemented loss functions:** Jiakai Shi, Mingxuan Teng.
- **Implemented evaluating metrics:** Jiakai Shi, Jiaming Yang.
- **Prepared datasets for training and evaluating results:** Jiaming Yang.
- **Trained, evaluated DRPI-CGAN model:** Jiakai Shi.
- **Analyzed results from evaluation:** Mingxuan Teng.
- **Visualized training and evaluation results:** Jiaming Yang.
- **Wrote the report:** Mingxuan Teng, Jiakai Shi, Jiaming Yang.

B.2 By Person

Mingxuan Teng developed the DRN structure in DRPI-CGAN infrastructure, designed and implemented adversarial loss functions, designed DRPI-CGAN architecture, analyzed the training loss results and evaluation results, wrote Abstract, Introduction, Experimental Results and Conclusion section in the report.

Jiakai Shi developed the PWC-Net structure and GAN structure in DRPI-CGAN infrastructure, tested the usability of DRN and PWC-Net models to generate optical flows from images, built data pipeline in Torch framework for training DRPI-CGAN model, designed and implemented end-to-end point error loss and warp loss functions, designed and implemented DRPI-CGAN architecture, implemented EPE and ΔPSNR evaluating metric, trained and evaluated DRPI-CGAN model, wrote Method section in the report.

Jiaming Yang implemented IoU evaluating metric, prepared and cleaned original and modified images from FFHQ Dataset, visualized and generated figures for training and evaluating results, wrote Reference and Appendix section in the report.