Let It Flow

Katayoon Goshvadi
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
kgoshvad@andrew.cmu.edu

Muhammad Suhail Saleem
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
msaleem2@andrew.cmu.edu

Akshay Sharma
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
akshaysh@andrew.cmu.edu

Abstract

Optical flow is an important concept that plays a central role in various computer vision tasks like object tracking and image segmentation. With the recent advancements in deep learning, especially convolutional neural networks (CNNs), there has been a surge of learning based flow estimation methods in an attempt to overcome the limitations of the classical approaches. Most of these learning-based methods use an autoencoder based architecture to generate optical flow from a given set of images. But to the best of our knowledge there has been no attempt at trying to tackle this task using more complex models like Variational Autoencoders (VAE) and Generative Adverserial Networks (GANs). Given the exceptional and robust performance the two deep learning models have delivered in recent years, especially on computer vision tasks, we have made an attempt to estimate optical flow using them. In this report we talk about our implementation of the models, analysis of their performance on the task and a discussion of the various challenges we faced while training them.

1 Introduction

Optical flow is defined as the apparent motion of individual pixels in the image plane and can be represented as a two-dimensional vector-field. This is an important and interesting problem as the estimation of optical flow is used in a number of areas including object tracking, autonomous driving, object segmentation, and video semantic understanding.

The problem of computing optical flow given two consecutive frames, in its classical formulation was tackled by making several assumptions about the flow and the frames. Some of the more common assumptions that have been used in the past literature include brightness constancy of the frames (i.e. the values of the pixels gradually changes between pixels and there are no significant jumps) and spatial smoothness of the flow (i.e. the value of the flow gradually changes between pixels and there are no significant jumps). These assumptions are only coarse approximations to reality and they limit the performance.

Recent approaches have tried to start over, by focussing more on the use of neural network based methods. While the initial focus was on supervised architectures, the lack of real-world ground truth data has forced people to develop unsupervised and self-supervised techniques. In our previous approach to the problem, as part of the project for the Introduction to Machine Learning for Engineers in the Fall of 2018, we had developed a self-supervised convolutional neural network framework, while accounting for temporal flow smoothing (i.e. the value of the flow of a pixel gradually changes between frames and there are no significant jumps). So instead of estimating optical flow between 2 consecutive frames, we used 3 consecutive frames to estimate 2 optical flows (one between frame 1 and frame 2, the other between frame 2 and frame 3). This way the estimation of both the optical flows make use of the information from all 3 frames.

In this project, we approach the problem of optical flow estimation between 2 frames, through the use of more complex models like Generative Adversarial Networks (GANs) and Variational AutoEncoders (VAEs). The reason for the shift being twofold:

- Both GANs and VAEs can give us robustness over noisy input (especially in terms of motion intensity)
- In the previous techniques we had to manually define loss functions for training the network. Most of these loss functions are based on the classical assumptions mentioned earlier (spatial smoothing and brightness constancy). This could cause the network to not capture some important features that could aid in the accuracy of the estimation process. By using GANs, we let the generator learn the features not from the loss functions based on these assumptions but instead through a discriminator. This allows the generator to learn a wide array of features not curbed by the assumptions.

2 Related Work

One of the primary methods of optical flow estimation that is currently being used in standard computer vision libraries like OpenCV[1] is Lucas-Kanade[8]. Belonging to the class of differential methods, it uses Taylor series approximation for estimation. Apart from the approximations, as mentioned earlier, belonging to the set of classical algorithms, it also assumes that the flow is constant between neighboring pixels. Although the technique produces good results, these assumptions limit the accuracy of the estimation. This was the primary motivation for using a deep learning based approach for optical flow estimation.

One of the more prominent works amongst the deep learning based approaches is FlowNet [2]. In this work they feed in a pair of consecutive frames from a synthetically generated dataset as input to a convolutional neural network and predict the optical flow between the frames. Since the frames are synthetically generated, ground truth optical flow was readily available and hence the model was directly trained by comparing the prediction with the ground truth.

FlowNet 2.0 [6] a paper succeeding FlowNet, discusses about combining two algorithms FLowNetS and FlowNetC together, while training them individually on synthetic datasets (in a specific order) so as to improve the accuracy. SpyNet(Optical Flow Estimation using a Spatial Pyramid Network)[12] discusses combining several individually trained networks as a spatial pyramid network. Here the flow is estimated at the highest level of the pyramid using low resolution images. The computed residual flow is passed onto the lower levels until we obtain the flow at the highest resolution. The major drawback of these approaches was the fact that they had used synthetic data which would significantly impact the accuracy of their model when tested on real world data. Furthermore, trying to combine several individual networks and training them independently on different datasets in different orders induces several parameters which requires extensive hyperparameter tuning.

An approach which overcomes the above mentioned drawbacks is UnFlow [10]. This is an unsupervised learning approach which predicts the optical flow when fed in two consecutive frames. This optical flow is then used to warp the first frame to obtain the predicted second frame. The model is then trained by comparing the predicted second frame with the ground truth second frame. This way, the training process is unsupervised and has eliminated the need for ground truth flows.

We have drawn inspiration from the above works in our attempt to predict optical flows using complex models.

3 Proposed Method

In this section we will discuss the details of our approach, the architectures of our proposed models (VAE and GAN) and the loss functions used for training them. To understand and compare the performance of our approach, we implemented two autoencoder based baselines. The first is a supervised autoencoder and the second is an unsupervised autoencoder. Details about each of these approaches can be found in the subsections below.

3.1 Supervised Autoencoder

FlownetC [3], one of the more prominent works in the field was chosen as the supervised autoencoder baseline. This is a supervised model that extracts optical flow at different scales, thereby learning strong multi-scale features.

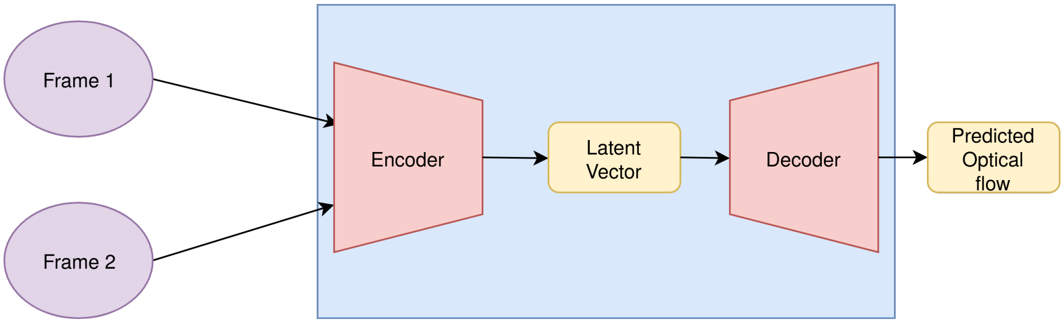


Figure 1: Supervised AutoEncoder Architecture

For every pair of consecutive frames, the optical flow at every scale is upscaled to the ground truth optical flow's size (Fig. 1), and the L_2 loss between the Flow$_{pred}$ and the Flow$_{true}$ is used to train the network.

$$L_{\text{AE}} = ||Flow_{pred} - Flow_{true}||_2 \tag{1}$$

3.2 Unsupervised Autoencoder

The unsupervised autoencoder that we used was inspired by the work of Meister et. al. [10]. Since the goal of this approach is to alleviate the problem of the requirement of ground truth flows to train the model, instead of comparing the flows, the predicted flow is warped with the first image to obtain a predicted second frame. This predicted second frame is compared with the ground truth second frame (which was part of the input) to train the model as shown in Fig. 2.

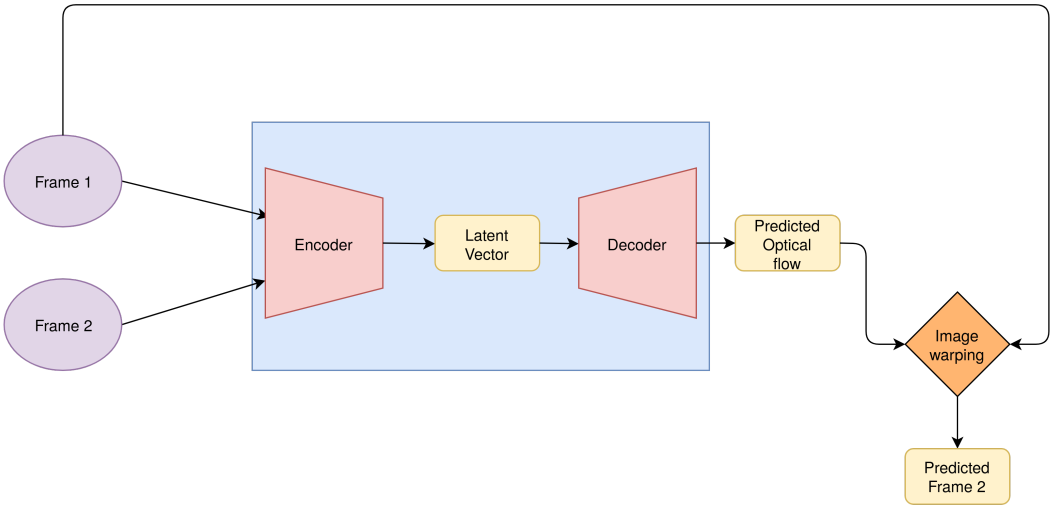


Figure 2: Unsupervised AutoEncoder Architecture

In an unsupervised setup it is common to add spatial smoothing loss over the Flow$_{pred}$ map. This stems from the classical assumption, that the optical flow of a small neighborhood of pixels should be the same. Classical setups like [9] apply it as a hard constraint enforcing this property to hold for all pixel neighborhood, whereas the current deep learning based methods like [10] use it as a penalty function which is added to the loss function.

$$L_{\text{UAE}} = ||\textit{Predicted Fame 2} - \textit{Frame 2}||_2 + L_{\text{Spatial smoothing}} \tag{2}$$

We calculate the smoothing loss using the Charbonnier loss function $\mathbf{C}(x)$ as described in [10].

$$\mathbf{C}(x) = (x^2 + \epsilon^2)^{\gamma} \tag{3}$$

where ϵ is a small number in the range $(0, 1)$, and γ is an exponent which is usually kept at 0.45.

$$L_{\text{Spatial smoothing}} = \sum_{\mathbf{x}} \mathbf{C}\left[\textit{Flow}_{pred}[N(\mathbf{x})] * F_1 + \textit{Flow}_{pred}[N(\mathbf{x})] * F_2 \right] \tag{4}$$

$$\tag{5}$$

where $N(\mathbf{x})$ defines the neighbourhood of the pixel location $\mathbf{x}$, F_1 and F_2 are the Sobel filters[13] for x and y image derivatives respectively, and $*$ is the convolution operator.

3.3 Unsupervised VAE

The major difference between an ordinary autoencoder and a variational autoencoder is that a VAE encodes the input data in a regularised probability space defined by a standard normal distribution, $\mathcal{N}(0, I)$ (Fig 3). This allows for more robust encoding of the inputs, and can thus lead to potentially better performance on test data. As explained in the unsupervised autoencoder method, the FlownetC [2] model extracts features at different scale to generate optical flow, we in particular modified the last encoded feature vector and embedded it into the probability distribution required for a VAE.

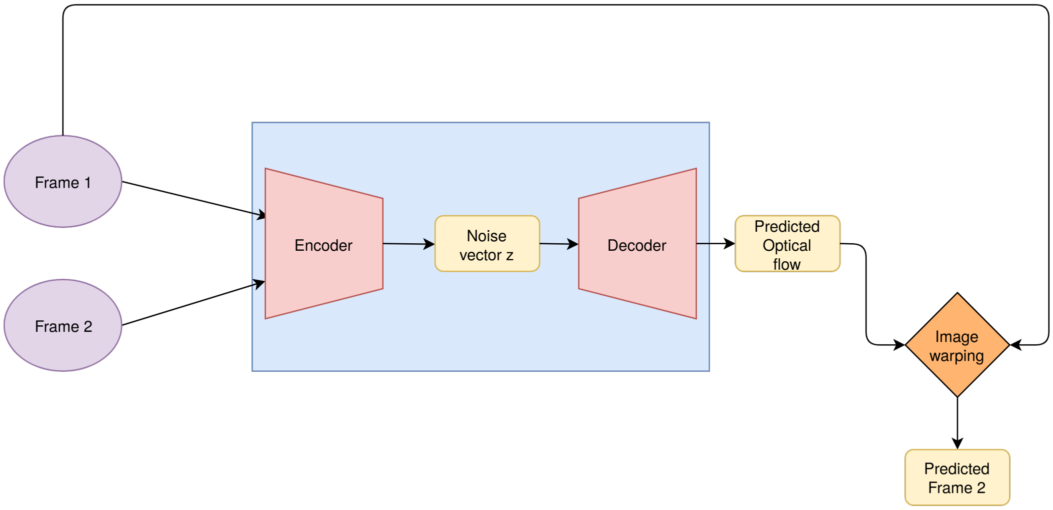


Figure 3: Unsupervised Variational AutoEncoder Architecture

The technique used to train the VAE is similar to the training process of the unsupervised Autoencoder, except that in this case we use an additional KL divergence loss term that is used to regularise the target encoding distribution.

$$L_{\text{VAE}} = KLD + ||\textit{Predicted Frame 2} - \textit{Frame 2}||_2 + L_{\text{Spatial smoothing}} \tag{6}$$

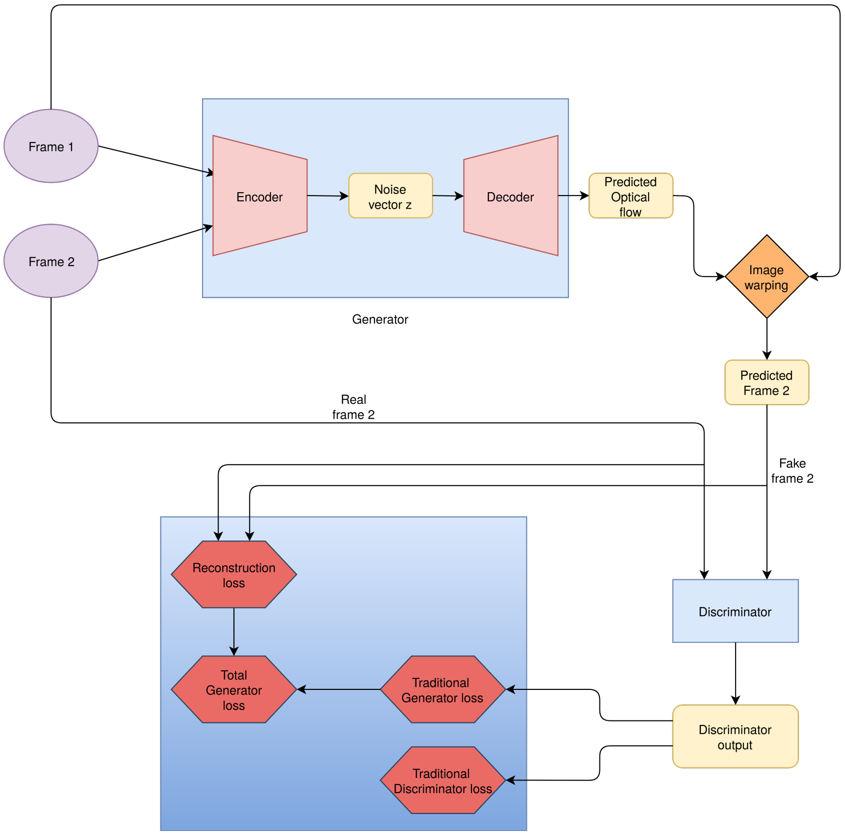


Figure 4: GAN Architecture

3.4 GAN

The final approach that we had implemented was a GAN[5] based architecture. We took inspiration from [14], which uses an input conditioned generator to perform semantic in-painting of a given image. Our architecture builds up on the previous unsupervised VAE based approach and instead of using a predefined loss functions based on the classical assumptions (like the smoothing losses), we pass both the ground truth frame 2 and the predicted frame 2 through a discriminator. The intuition behind this is that the discriminator will slowly learn the finer details present in the ground truth frame 2 and thus will force the generator to generate those details through the optical flow prediction. This way there will be no need to explicitly encode the assumptions. The framework of our proposed model is shown in Fig 4.

Generally in a GAN architecture, to generate the output we randomly sample noise and feed that to the generator. However, we want the prediction made by the GAN to be dependent on the input frames and not any random optical flow. So we would like the noise (or the latent vector) to be conditioned on the input, so that the output corresponds to the inputs. For this reason the first part of our generator is an encoder which learns to map the input frames to a standard normal distribution similar to a VAE. The second part would be the decoder, which generates an optical flow according to the noise sampled from this distribution and is used to warp the **Frame 1** to obtain **Predicted Frame 2**. Now this **Predicted Frame 2** and **Frame 2** are passed on to the discriminator as fake and real data respectively. The network is trained using a combination of traditional GAN losses[5] and the KL divergence loss on the probability distribution, for the VAE part in the generator.

Generator Loss:

$$L_{\text{Gen}} = -\log(Disc(\textit{Predicted Frame 2})) + KLD + L_{\text{Spatial smoothing}} \tag{7}$$

Discriminator Loss:

$$L_{\text{Disc}} = -\log(Disc(\textit{Frame 2})) - \log(1 - Disc(\textit{Predicted Frame 2})) \tag{8}$$

4 Experiments

4.1 Datasets

We focused on two main datasets for this task: KITTI flow 2015 dataset [11] and the MCL-V dataset [7]

4.1.1 KITTI Dataset

The KITTI dataset is a very popular dataset used to benchmark optical flow estimation methods. This dataset consists of first person view from a car being driven in different urban scenarios and highways. This makes this data a very good choice from the viewpoint of real world autonomous car scenario, which does depend on a good optical flow estimator to get real time velocities of objects around it.

4.1.2 MCL-V Dataset

The MCL-V dataset has videos of both animated and real world scenes. Both these setting have a lot of varieties in terms of motion directions, and intensities, which can provide a lot of robustness to networks trained on it.

4.2 Evaluation Metric

To evaluate the performance of the models, we use what is referred to as End Point Error (EPE) as the evaluation metric. EPE is a standard metric used to judge the quality of the Flow$_{pred}$, and is calculated as the euclidean distance between the Flow$_{pred}$ and the Flow$_{true}$. The ground truth flows for some sequences in KITTI dataset [11] are readily available, hence the EPE is computed with respect to these flows.

$$EPE = ||Flow_{pred} - Flow_{true}||_2 \tag{9}$$

4.3 Results and Discussions

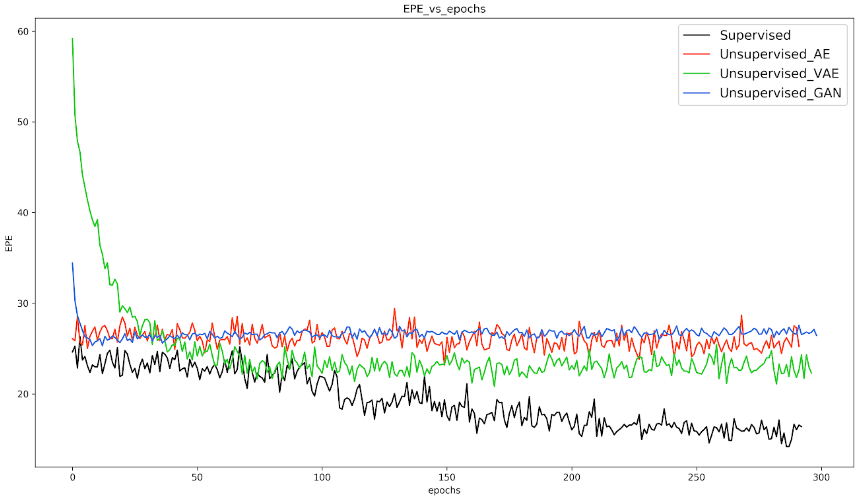


Figure 5: EPE comparision for all the methods

We ran all the models on the training split of the KITTI[11] dataset for 300 epochs and plotted the EPE for every 10^{th} epoch on the testing split of the dataset. Fig 5 shows the EPE vs Epoch plot for all the models discussed in section 3. According to the plot, the supervised model was able to achieve the lowest EPE of all the other models, with the GAN performing the worst. This is contrary to the hypothesis we started out with.

The Flow$_{pred}$ and the warped frame 2 for each model can be seen in Fig 6. As can be seen, the supervised autoencoder and the unsupervised autoencoder return acceptable and smooth flow predictions. However, the VAE returns an extremely noisy output resulting in a distorted second frame. On the other hand, GAN failed to learn a useful prediction as it continued to predict flow values closer to zero. These results do agree with the EPE plots for each model, where the supervised autoencoder performed the best.

It has to be kept in mind that this is a research problem that requires a lot more time to analyse the cause of the issues and develop solutions. However, in the limited time available to us, we ran a lot of experiments to understand more about the possible reasons for the models' failures. Some of them are listed below:

- **Reconstruction Loss:** We had tested the impact of both L1 and L2 losses on the training of the model. Neither of these resulted in favorable results.
- **Network weight initialization:** For both the GAN and VAE setups. We noticed that the KLD loss was blowing up to very high values very quickly. On deeper analysis this pointed towards the problem being the initialization of the network weights. We found that initializing the network weights using Xavier[4] normal initialization between $[-0.1, 0.1]$ sets this issue right and does not lead to the blow up of the KLD loss.
- **Spatial smoothing loss:** For the GAN based model we were initially not considering any smoothing loss over the optical flow, as we were working under the assumption that the learning signals from the discriminator will help the generator learn these features without explicitly encoding them. Since this was not the case, we had also tried explicitly encoding the spatial smoothing through a smoothing loss function. However, this did not change the results.
- **Architectures for VAE and GAN:** Before switching to the Flownet-C[3] architecture, we had tried a custom architecture for the VAE and GAN model. This architecture did not use the multi-scale optical flow estimation introduced by Flownet-C. We were hoping the use of a discriminator will avoid the requirement for a multi-scale prediction. However, this was not the case. Hence we switched to a multi-scale predictor. Even after the switch, there were no visible improvements in the performance. The implementations of both these models can be found in the section below .

5 Implementation

Here is a link to the 2 repositories that we had developed our code in. The first repository was developed from scratch and contains the code for the GAN and the VAE models while the second repo is a fork from FlowNet [2] that we added our GAN and VAE implementation to.

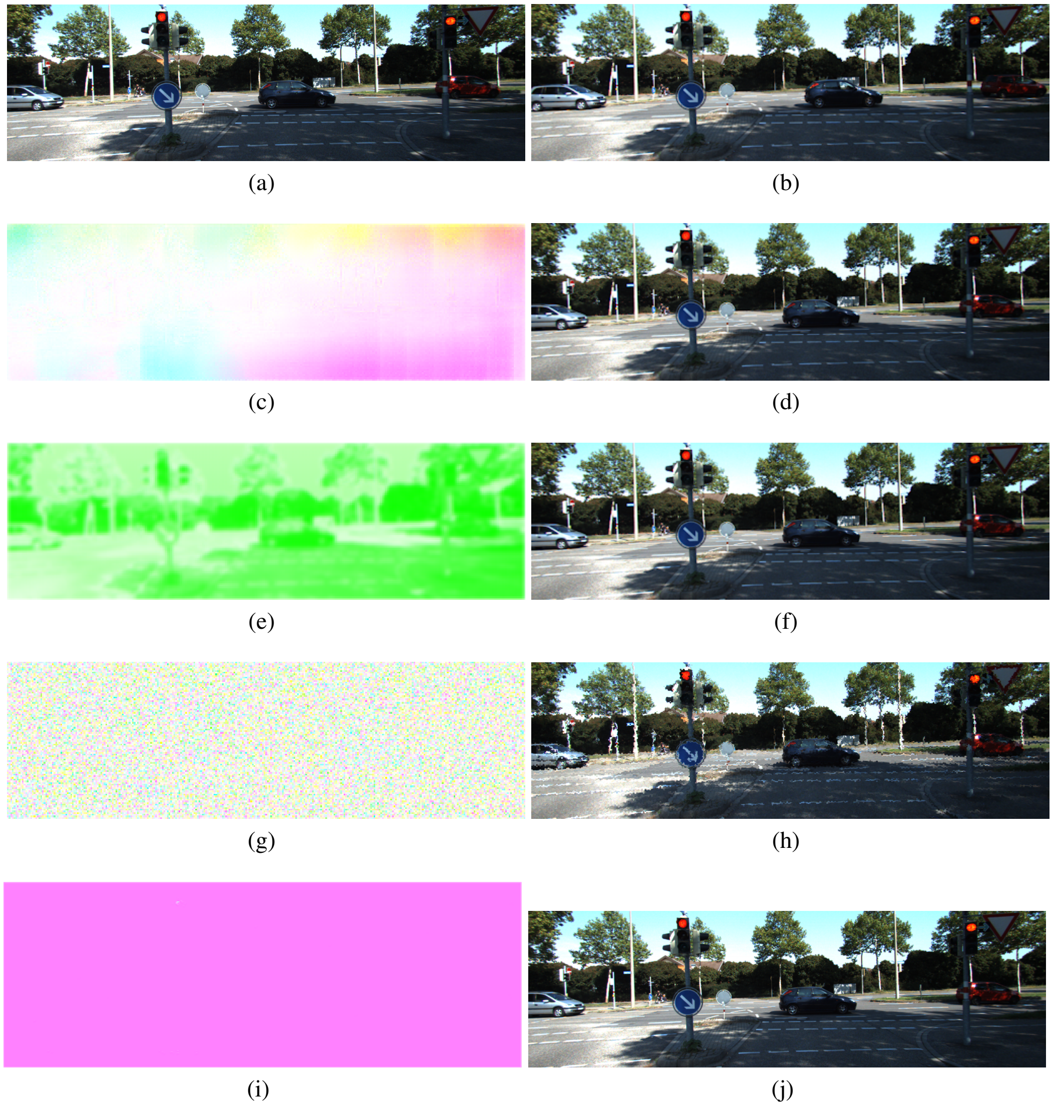
- **Original implementation:** https://github.com/akshay-sharma1995/let_it_flow.git (master branch)
- **Flownet-C based implementation:** https://github.com/Muhammad441/FlowNetPytorch.git (unsupervised model branch)

6 Conclusion and future work

In conclusion, we have made an attempt to use complex models like GANs and VAEs to predict the optical flow between consecutive frames (although unsuccessfully). One of the major drawbacks of using complex models like GANs, is that training them can be tricky and might require a lot of engineering and analysis. While we have tried to analyse a fair share of the issues, the problem warrants more time and effort which we hope to invest post-deadline.

7 Acknowledgement

We would like to thank Dr. Farimani and the course TAs for providing insights into the working of the various deep neural network architectures.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 6: (a) Ground truth frame 1 (b) Ground truth frame 2 (c) Supervised AE predicted flow (d) Supervised AE predicted frame 2 (e) Unsupervised AE predicted flow (f) Unsupervised AE predicted frame 2 (g) VAE predicted flow (h) VAE predicted frame 2 (i) GAN predicted flow (j) GAN predicted frame 2

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[3] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. apr 2015.

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

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

[6] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:1647–1655, 2017.

[7] Joe Yuchieh Lin, Rui Song, Chi-Hao Wu, TsungJung Liu, Haiqiang Wang, and C-C Jay Kuo. Mcl-v: A streaming video quality assessment database. *Journal of Visual Communication and Image Representation*, 30:1–9, 2015.

[8] Bruce D. Lucas and Takeo Kanade. Iterative Image Registration Technique With an Application To Stereo Vision. 2:674–679, 1981.

[9] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[10] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 7251–7259, 2018.

[11] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[12] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 2. IEEE, 2017.

[13] Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.

[14] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with deep generative models. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:6882–6890, jul 2017.